THE SECANT AND TRAVELING ARTIFICIAL POTENTIAL FIELD APPROACHES TO HIGH DIMENSIONAL ROBOTIC PATH PLANNING

A Dissertation
Presented to
The Academic Faculty

By

Konrad J.H. Ahlin

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mechanical Engineering

Georgia Institute of Technology

December 2018

THE SECANT AND TRAVELING ARTIFICIAL POTENTIAL FIELD APPROACHES TO HIGH DIMENSIONAL ROBOTIC PATH PLANNING

Approved by:

Dr. Nader Sadegh, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Ai-Ping Hu
Institute for Robotics and Intelligent Machines
Georgia Tech Research Institute

Dr. Jun Ueda
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Hao-Min Zhou
School of Mathematics
Georgia Institute of Technology

Dr. Volkan Isler
Department of Computer Science
University of Minnesota

Date Approved: August 24, 2018

The true delight is in the finding out rather than in the knowing.

Isaac Asimov

ACKNOWLEDGEMENTS

The journey this thesis has taken me on will shape my life, and I do not have the words to properly thank those who have helped me along the way. From the day-to-day struggle to the carefully followed plans, my wife, friends, and mentors have made my work possible.

Thank you to Dr. Sadegh and Dr. Hu. Your guidance and insight has made this work possible. We have taken some twists and turns in getting this thesis to where it needed to go, but I could not be happier with the outcome. Hopefully, this is only the first chapter to an even greater story, but the lessons I have learned will help to guide me through the rest of my career.

Thank you to the people at GTRI. Although there are too many to list, you have all been with me as I fought my way through the daily trials of research and discovery. Your support has made the long journey feel ephemeral, and in working with you all, I have shaped and grown in my professional life as well. I look forward to the work that is still left to do.

Finally and not least, thank you Katie. No matter how long the days lasted, no matter how bad Atlanta traffic might be, every day I could come home to you. You have been by my side through every step, and this is just one of the adventures we will travel together.

Oh, and Joe, thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The field of robotic path planning is rich and diverse. As more complicated systems have become automated, the need for simple methods that can navigate high dimensional spaces has increased. However, most path planning methods, such as Road Map methods and Search methods, increase exponentially with dimension, making them undesirable for complex robotics. Thus, the Secant and Traveling Artificial Potential Field (TAPF) approaches were developed. The Secant and TAPF approaches are modifications to the general Artificial Potential Field (APF) path planning algorithm with desirable properties, which make them ideal for path planning in high dimensional space. All APF methods grow linearly with dimension; however, general APF methods are not guaranteed to converge given an arbitrary field of obstacles, significantly hindering the applicability of the APF algorithm. By specially tuning the artificial forces generated by the Secant and TAPF approaches, these methods can be shown to be globally asymptotically stable at the target location for a point robot in a field of point obstacles. To extend this theory for more practical applications, the concept of a boundary layer was introduced into the path planning algorithm. The boundary layer is a finite radius that encompasses an obstacle, such that the field is transformed within the boundary layer to account for the solid shape. By warping the landscape within the boundary layer, the system becomes mathematically equivalent to avoiding a point in space. From these advancements, the Secant and TAPF approaches were then demonstrated on planar robots and manipulators. These real-world systems were handled by selecting individual points on the robot that need to converge and treating them as separate systems coupled together by the defined constraints. For example, a planar robot is dynamically equivalent to two points constrained by a link. Similarly, a manipulator could be considered to be n-points jointed together. With the use of the Secant and TAPF approaches to the APF algorithm, robotic control and path planning could be drastically simplified, even for complex systems.

CHAPTER 1
BACKGROUND INFORMATION

1.1 Introduction

Robotic path planning is extremely diverse, and countless methods and theories have been developed over the past half century to run autonomous systems. One such theory, Artificial Potential Fields (APF), approaches path planning in a way that makes it advantageous for operating complex systems. Despite the benefits, the theory has limitations that make it difficult to implement and it is no longer widely popular. However, by adjusting the governing equations using new methods, which will be called the Secant and Traveling Artificial Potential Field (TAPF) approaches, many of the issues facing the general APF theory can be addressed. The Secant and TAPF approaches are generalizable path planning methods which could be used to control many different forms of robotic systems with a simple and intuitive nature.

As the complexity of robotic systems increases, the process of path planning to avoid obstacles and converge on targets becomes more challenging. Typical path planning methods included a wide variety of approaches, such as: road map, cell decomposition, APF, sampling based, and optimal methods. Each of these approaches have their appropriate applications, but any conceivable method will in some way be dependent on the dimension of the space that needs to be navigated. Current methods typically try and improve the efficiency of path planning, but this does not address the fundamental issue that arises with increased dimensionality. As a result, path planning for higher order systems can be challenging and computationally expensive. Of all the path planning methods mentioned, only APF offers the potential for a simple solution: develop a routine that scales linearly with dimension. Search methods, whether they require global information or not, require ex-

ponential growth with dimension, no matter how efficient they are otherwise. Conversely, APF routines inherently grow linearly with dimension. As systems become more complex, the need for a simple and yet sophisticated path planning algorithm grows.

Artificial Potential Field theory is an intuitive form of path planning. The method creates a fictitious attractive force at the desired position and repellent forces at obstacles. The summation of forces for any given point creates a potential field from which the theory receives its name. The path planning is derived by simulating the behavior of the artificial dynamic system and using the results as a basis of control for the actual system. This method has a level of simplicity and elegance which makes it a powerful tool. Unfortunately, the method heavily suffers from issues with convergence. The primary flaw with the general method is its propensity to get trapped in local minima. In the potential field, if the gradient at a given point decreases to zero with a positive second derivative, the robotic system could converge to this point rather than goal position. Numerous methods over the past 30 years have been developed to augment the APF method and reduce the possibility of a local minimum. Many of these methods include combining the APF method with other path planning techniques, but this chains the APF approach with the dimensional growth of other algorithms. While various APF routines may work in specific scenarios or with contrived conditional parameters, the Secant and TAPF approaches will be shown to be generalizable path planning methods.

The Secant and TAPF approaches will demonstrate that Artificial Potential Field theory could be modified to account for local minimum issues. Typically, the convergence of APF methods will depend on the placement of obstacles within the field. As a result, if the system becomes trapped in a local minimum, it cannot recover on its own. However, by carefully crafting the potential field and the resultant gradient, the Secant and TAPF approaches manage to be globally convergent and generalizable to any field of obstacles. APF methods typically rely on well tuned gain parameters to avoid local minimum, which often depends on the position of obstacles relative to one another. The Secant and TAPF

approaches do not need to consider the relative position of obstacles in order to guarantee convergence. These methods are extremely robust in trajectory planning in n-dimensional space.

The trap of path planning is to make an algorithm which is only applicable to a specific system. Often, in trying to boost performance, modifications will make a theory less generalizable. APF theory benefits from being fundamentally simple while also extremely malleable. This analysis will begin by examining point robots in n-dimensional space with point obstacles; however, it will then be expanded to solid bodied obstacles and then practical applications. The Secant and TAPF methods have the possibility to be applicable to various forms of robotic systems of high complexity while maintaining necessary convergence reliability.

1.2 Motivation

High dimensional robotic path planning is necessary to modern applications. Robotic manipulators in particular are extremely difficult to design suitable path planning routines. High dimensional systems are typical for robotic arms, and the transformation from task space to configuration space is nontrivial. Furthermore, finding a path planning routine robust enough for generalized robotic arms is quite difficult: manipulators can be any size or dimension operating in arbitrary environments. The Secant and TAPF approaches were developed to apply to robotic arms, but their characteristics and limitations will first be described in n-dimensional space for point robots and obstacles. Then, the systems will be naturally expanded for real world and practical environments. First, however, the field of path planning routines will be examined in brief in order to show the need for the Secant and TAPF approaches.

1.3 Literature Review

1.3.1 Path Planning Overview

Methods and techniques have developed within path planning naturally over the span of half a century; adapting to changing requirements and capabilities. Algorithms are interrelated and interdependent. Specializations are developed for individual applications. Hybrid algorithms are common. Techniques will wax and wain in popularity and perhaps fade into obscurity. Different authors have differing classifications of these algorithms that may or may not be fully descriptive of their capabilities and requirements. This summary will attempt to capture the trend and development of these methods with the understanding that it is not complete, but it will at least be descriptive of the developments in the field. This summary is limited to methods that are generally applicable to robots with various constraints rather than specific robotic designs with specialized considerations.

1. Road Map Methods
    (a) Visibility Graph
    (b) Voronoi Diagram
    (c) Freeway Net
    (d) Silhouette
2. Cell Decomposition
    (a) Exact Decomposition
    (b) Approximate Decomposition
3. Potential Field Approach
    (a) General Artificial Potential Field
    (b) Harmonic Potential Field

4. Sampling Based Methods

    (a) Probabilistic Road Map (PRM)

    (b) Rapidly Exploring Random Trees (RRT)

5. Optimal Searches

    (a) A*

    (b) Covariant Hamiltonian Optimization Map Planning (CHOMP)

    (c) Simultaneous Localization and Mapping (SLAM)

The ordering of the list shown in Section 1.3.1 is deliberate and roughly tracks the history of path planning methods over the past half-century [1] [2] [3]. Robotic path planning is traditionally constrained by a robot's ability to sense its environment and the time required to compute or recompute viable paths. As sensor performance and computational power has increased, these methods have become more sophisticated. However, robot design complexity has also increased, creating an arms-race for algorithms to be simultaneously computationally efficient while remaining sufficiently powerful to navigate an n-dimensional device. Path planning approaches are typically separated into two categories: local and global methods [4] [5] . Local methods only consider the immediate environment while global methods account for the entire space [6] [7]. However, this research so far has uncovered another set of categories that path planning methods usually subscribe to: mapping and minimization. Mapping methods (such as cell decomposition) consider how to divide a space into navigable parts; minimizing methods (such as A*) consider the sequence of navigable parts necessary to reach the goal and avoid obstacles. Often complete methods will blend global and local methods as well as mapping and minimization methods. The duality of mapping methods and minimizing methods is mentioned here because the classification was not found within the research. The above list of methods is too numerous and the algorithms are too similar to examine each in detail, so only a collection of relevant algorithms will be examined, listed in Table 1.1.

Table 1.1: List of algorithms for examination.

Class	Algorithm	Application	Domain
Road Map Method	Voronoi Diagrams	Mapping	Global
Cell Decomposition	Approx. and Exact	Mapping	Global
Sampling Based Methods	RRT	Mapping	Local\Global
Optimization	A*	Minimizing	Local\Global
Optimization	CHOMP	Minimizing	Local
Artificial Potential Fields	General APF	Minimizing	Local

This research is focused on examining and improving the Artificial Potential Field method in order to develop an algorithm that is applicable to high dimensional systems and generalizable to a variety of robotic designs. However, the justification of this research is based on the existing landscape of path planning methods; their limitations and their strengths. Thus, the above methods will each be examined in brief and then the APF method will be examined more thoroughly so that an understanding of the challenges surrounding path planning can be understood in full.

1.3.2 Voronoi Diagrams

Method: Path planning via Voronoi diagrams usually occur in three steps: construct the generalized Voronoi cells, compute the retractions on the configuration space that include the start and goal positions, and then search the resulting free space for connecting paths from the start position to the goal position. Voronoi cells isolate obstacles in individual spaces that maximize the available space surrounding each obstacle. The retraction process limits the navigable space to the boundary of the Voronoi cells, thus ensuring that the resultant pahts are free of obstacles. Voronoi diagrams rely on global information in order to successfully map the space [8] [9].

Benefits: Voronoi diagrams are guaranteed to show the free path from any given start position to any desired goal position as long as a free path exists. Voronoi diagrams are like most road map methods in that they attempt to limit the available paths into a subset that

are guaranteed to be free of obstacles [10].

Detriments: Voronoi diagrams do not provide a method for choosing the appropriate path from the start position to the goal. They are a mapping method which rely on a secondary minimization method to converge on the goal position [11]. Voronoi diagrams are also specifically chosen because they maximize the distance between obstacle locations, which is not always beneficial and can lead to non-optimal routes [12]. Also, for large spaces with numerous obstacles (and dimensions greater than 2), Voronoi diagrams can quickly escalate in computational cost.

1.3.3 Cell Decomposition

Method: Cell decomposition (both approximate and exact) attempt to subdivide the workspace into feasible and infeasible regions. Unlike roadmap methods, this does not involve a retraction of the space, rather it simply categorizes the space based on the location of the obstacles. Like roadmap methods, this is also a mapping technique, which requires global information [8] [12]. The difference between approximate and exact cell decomposition is that the exact method will create cells of arbitrary space necessary to fully define the workspace. Approximate cell decomposition will use predetermined cell shapes (usually rectangular) and categorize the space to within a certain tolerance.

Benefits: Cell decomposition is guaranteed to connect the target position with the goal position so long as a viable path exists. It also does not require any retraction of the workspace, but rather wholly defines the free space within an environment [13].

Detriments: Like Voronoi diagrams, cell decomposition does not inherently define a suitable path. Also, while cell decomposition retains more information about a space over retraction methods, this can at times make finding optimal paths more difficult [14]. Also, like all global methods, cell decomposition becomes much more costly for higher dimensional spaces.

1.3.4 RRT

Method: RRT is a sampling based approach that provides a thorough way of both searching and navigating a space. The RRT method constructs a series of connected nodes that define the free space within the environment. By adding, branching, and connecting nodes to existing vertices, RRT expands the known free space of the environment to encompass the entire space [15] [16] [17]. With consistent checks, the planner can detect if the goal and start positions have become linked by the network of nodes. Sampling based methods offer a unique approach to the path planning dualities of mapping and minimization as well as to local and global planning. The root of the method is inherently a local mapping problem. Collisions are not detected until a new node is added to the branching network: thus the obstacle does not need to be detected until it is local to the search path [18]. Also, the primary objective of the algorithm is to map the space without regards to seeking the goal. However, by combining the local information throughout the entire workspace, the method becomes a global search. Thus, by efficiently mapping the space, information about the minimum path to the goal can be obtained.

Benefits: RRT is a robust path planning algorithm that is often used in the robotics community. The method is efficient at finding global convergence while only needing to consider local information, making it ideal for simulating feasible paths [19].

Detriments: RRT is not practical in applications where the environment is unknown a priori. The number of iterations that are required to map the space make experimental implementation of the method nearly impossible. Instead, the algorithm is appropriate for simulations, where the route is discovered before experimental applications. Also, while the algorithm can find a path from the start to the goal, this path may be sub-optimal, and many algorithms will apply an optimization routine after the RRT method is utilized to smooth the path. Finally, RRT suffers in dense, large spaces and the search time increases exponentially with increased dimensionality [20] [21].

1.3.5 A*

Method: The A* approach is one of the most versatile and widely used algorithm in robotic path planning. The method balances a set of open and closed nodes and decides which node to test in attempting to converge towards the target. A* uses a cost function that it works to minimize; allowing for the algorithm to not only find a suitable path, but to find the best path possible [22] [23]. By constantly exploring the boundary of the closed set, the method is guaranteed to converge on the target location while avoiding obstacles. A* is purely a minimization method and not a mapping algorithm. The routine relies on the space to already be defined by a series of classified and connected nodes. Often times, this step is accomplished by using approximate cell decomposition (creating a grid of the space with obstacles identified), but more sophisticated or targeted methods are also applicable [24]. Similar to RRT, A* is a local method that grows into a global method. Nodes only need to be evaluated when the algorithm encounters them. Unlike RRT, A* is less likely to explore irrelevant space. While RRT attempts to "fill" the free space, A* simply identifies the paths in the direction of its minimization. Often times, A* will be combined with mapping methods, such as RRT, to create a suitable hybrid approach that efficiently maps the space and minimizes relevant parameters in approaching the goal position.

Benefits: A* has long been utilized by the robotics community because of its versatility. The method does not have an inherent mapping component, but the algorithm is strong enough that even a simple mapping routine will suffice to provide excellent results [25]. A* can even account for additional parameters within its minimization routine to find optimal paths beyond just the shortest distance to the target [26].

Detriments: While listed as an optimal search within this research, A* has elements of a sample based method as well. The algorithm needs to account for a set of closed and open nodes, which it maintains to find the globally convergent path. However, this virtually requires the algorithm to be performed in simulation before it can be experimentally implemented. While modifications to the routine can be made so that simple experiments

can perform A* on-the-fly, these attempts are not guaranteed to succeed and may cause the need for back-tracking and inefficient motions, precisely what A* attempts to avoid. Also, since the routine operates by sampling nodes, it is hampered by increased dimensionality [27].

1.3.6 CHOMP

Method: The Covariant Hamiltonian Optimization for Motion Planning (CHOMP) routine is one of the more modern path planning algorithms in robotics. Its success has spawned a number of derivative and tangential methods, such as the Stochastic Trajectory Optimization for Motion Planning (STOMP). At its core, CHOMP is a trajectory optimization method often used for path planning of higher order systems, including manipulators and legged vehicles [28] [29]. The algorithm takes an initial naive trajectory to a target location and optimizes the path until it achieves desirable motion. By taking collisions with obstacles as a cost within the optimization function, CHOMP is able to find paths in high order space without the need of robustly searching the entire workspace. Trajectory optimization methods have existed beforehand, but CHOMP created a reliable algorithm for path planning with complex systems. CHOMP is a minimization technique, but it does not inherently rely on a mapping of the workspace to function (unlike A*). Also, CHOMP is a local routine, only considering the free space and obstacles immediately pertaining to a given position.

Benefits: The primary strength of CHOMP is its ability to handle high degree of freedom systems. Unlike sampling based methods, CHOMP does not require a search of the space it inhabits, rather relying on derivative information in order to optimize the trajectory. So, although the optimization still occurs within the same dimension as that of the system, the full configuration space does not need to be searched [30].

Detriments: Since CHOMP is an optimization routine, the algorithm can suffer from local minimum. Although CHOMP is designed to try and escape local minimum situa-

tions, researchers have found that the performance of the routine still sometimes suffers. Also, derivative information has to be available or numerically determined in order for the optimization function to operate. For these reasons, off-shoot methods such as STOMP were developed, each with their own strengths and flaws, but all following the example of trajectory optimization set by CHOMP [31] [32]. In order to converge, methods such as CHOMP still require gradient information, while methods such as STOMP require search techniques, both of which scale drastically with dimension. Finally, as previously stated, CHOMP is a local function, which inherently means that it is not guaranteed to converge on a solution, even if one exists.

1.3.7 Artificial Potential Fields

Method: The Artificial Potential Field approach has been utilized for decades in robotic path planning and has numerous variations to the algorithm. However, the underlying basis of the general APF method is to interpret goal positions as attractive forces and obstacles as repellent forces; the summation of which creates a potential field [33] [34]. Some methods will use the gradient of the potential field to describe the resultant velocity or acceleration rather than an applied force, but this derivation does not impugn the spirit of the approach. The APF method is a local minimization algorithm. However, like CHOMP, APF does not rely on a mapping of the environment into nodes, grids, or retracted free spaces. APF simply needs to know the location of obstacles and the goal position [35] [36] . Unfortunately, although APF has many benefits, the detriments have often outweighed the algorithm's usefulness, as will be discussed [37]. To compensate, a host of hybrid and alternate methods have been developed. Often, APF will be paired with a global mapping method that reduces the task space, such as Voronoi diagrams. However, this imposes limitations on the APF method which retracts from the algorithm's overall virtue. An alternative is to attempt to modify the method to improve performance. One of the most well known variations to APF is Harmonic Potential Fields. Harmonic Potential Fields uses a combination of source and

sink type fields which satisfy Laplace's equations. The method models obstacles as a series of lines that each provides a potential source. By adding a flow function and a sink function for the target, the method boasts the elimination of local minimums [38]. However, Harmonic Potential Fields cannot guarantee convergence [39]. The APF approach is simple, elegant, and powerful. Unfortunately, the problems with the theory usually outweigh the performance.

Benefits: One of the greatest aspects of the APF approach is the speed of the algorithm. The method does not rely on searches or optimizations, which means that the method does not suffer as greatly as others when applied in high dimensional spaces. All the method needs to compute is the summation of forces applied to the robot and the resultant state variables. The elegance of this theory is that the acting forces are inherent to the task space, and thus path planning can also be relegated to the task space domain. The described motion is derived by the transformation of the force into the configuration space via the system dynamics. In this way, the state values are informed by the result of the forces acting on the system. APF can utilize dynamics, which will increase with the degrees of freedom of the system, but which will not increase exponentially. Furthermore, since APF is a local method, the algorithm does not need to know the location of the obstacles until the robot has to avoid them [40]. Thus, APF is better suited for path planning of robotic systems, even those with high dimensional degrees of freedom in arbitrary environments.

Detriments: General APF theory has four, major, well documented problems:

1. Local minimum
2. No passage between closely spaced obstacles
3. Oscillations in narrow passages
4. Unreachable goal position in close proximity with obstacles

Of these issues, the local minimum problem is perhaps the most severe and the best known [41]. In brief, local minimums can result when the potential field has a zero gradient

and positive second derivative. In this instance, the algorithm may fail to converge on the target location [42]. Attempts have been made to avoid or eliminate this problem (such as Harmonic Potential Fields), but in general this problem infects all of APF theory. The remaining issues are minor in comparison with local minimum, but significant in their own right [43] [44]. If two obstacles are close together, the resulting potential field could disallow passage between them. In a similar way, if oscillations begin when traveling within a narrow passage, these oscillations will continue and could produce undesirable motion. Finally, an often overlooked artifact of APF methods: the goal position is unlikely the global minimum of the potential field, and this discrepancy becomes plain if an obstacle is close to the goal. Sometimes methods will attempt to account for local minimum within the algorithm and alter the approach [45], or contrive instances without obstacles [46], but this does not eliminate the underlying issue of local minimums. All of these issues make the APF theory nonviable in most path planning scenarios. The issues simply outweigh the benefits in all but the simplest applications, in which case other methods would be better suited. However, if a variant of the APF method could eliminate these issues while preserving the virtues of the algorithm, it could be extraordinarily powerful.

1.4 Statement of Work

With the understanding that a need exists for robust, high-dimensional path planning, the following work was completed and will be analyzed in subsequent chapters. This work is intended to demonstrate and analyze new forms of APF path planning: the Secant and TAPF approaches, which have desirable convergent properties. Chapters 2 and 3 introduce the Secant and TAPF approaches respectively and analyze the systems in n-dimensional space as point robots in a field of point obstacles. Then, Chapter 4 extends the theories to encompass arbitrarily shaped obstacles in n-dimensional space. The performance of APF methods is then compared to search algorithms in Chapter 5, with the intention of demonstrating the nature of dimensional growth. Finally, the methods are then adapted

to practical applications: planar motion with rotation in Chapter 6 and manipulators in Chapter 7. The summation of work will show that the Secant and TAPF approaches are extremely robust methods of path planning that can be utilized in general applications.

CHAPTER 2
THE SECANT APPROACH

2.1 Introduction

The Secant approach to APF path planning introduces a technique of constructing potential fields which will be applicable regardless of the configuration or number of obstacles to be avoided. To begin, this analysis will focus on point robots in n-dimensional space with point obstacles in the field. However the obstacles might be arranged, the Secant potential field has a global minimum at the target location. The gradient of this potential is then used in a control law which is provably, globally asymptotically stable at the target. These properties allow the Secant approach to be robust in its application.

As discussed in Chapter 1, the primary flaw with general APF methods is that they are subject to local minimum, causing the control law to converge on a point other than the target location. This local minimum occurs when obstacle configurations are such that the gradient of the potential field is zero outside of the target location. Typically, APF methods will attempt to alter their properties under these circumstances in order to create a more desirable gradient. However, as will be proven, the Secant approach creates a potential field which is incapable of having a gradient of zero anywhere accept at the target location. The ramification of this property is that once the field is constructed, it does not need to be modified to guarantee convergence.

The case of solving for point robots in n-dimensional space is not meant to be a practical application of the theory. However, in order to understand the underlying properties of the potential field and control law, this circumstance will be examined in depth. Further transforms and derivations to the theory will need to be developed in order to apply the Secant approach in practical applications. Understanding how the theory behaves in its

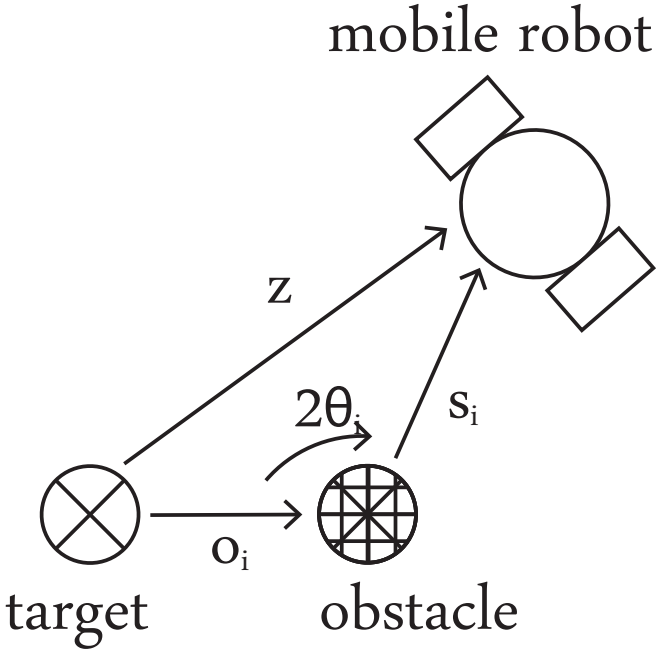


Figure 2.1: An example of a robot in the proximity of a target and an obstacle. Although the robot is depicted having wheels, it is assumed to be holonomic for the purpose of analysis. Furthermore, the target is the origin of the field; the vector z is the position of the robot in relation to the target, the vector o_i is the obstacle position, and the vector s_i is the vector between the obstacle and the target.

most basic form will be crucial in understanding how to utilize the theory in more complex applications.

2.2 Potential Field

As previously stated, this analysis will assume that the robot is a point in n-dimensional space. Figure 2.1 demonstrates an example of the field with the appropriate vector notations. Although Figure 2.1 is drawn with solid bodies, this is purely a graphical representation. The field will be constructed with a finite number of obstacles, the position of each of which will be represented by the vector o_i. The target will be assumed to be the origin of the field, and the position of the robot will be z with the relative vector to each

obstacle s_i. The angle formed by the position of the obstacle and the relative position of the robot is measured by 2θ. With these relationships, the potential function, $\Psi(z)$, for the Secant approach is shown in Equation 2.1. In this equation, k_p is the positive gain value on the attractive force, and k_i are a series of positive gain values for the repellent force. The mathematical derivation of the angle θ is demonstrated in Equation 2.2.

$$\Psi(z) = \frac{k_p}{2} z^T z + \frac{1}{2} \sum_i k_i \frac{z^T z \sec^2 \theta_i}{\|s_i\|^2} \tag{2.1}$$

$$\theta_i = \frac{1}{2} \cos^{-1} \frac{-(s_i)^T o_i}{\|s_i\| \, \|o_i\|} \tag{2.2}$$

The potential function in Equation 2.1 is guaranteed to have a global minimum at the origin of the field and to be devoid of other local minimums. The function is comprised of an attractive potential and a repulsive potential: $\Psi(z) = \Psi_a(z) + \Psi_r(z)$. The attractive potential, $\Psi_a(z)$, is purely a function of the distance to the target. The repulsive potential, $\Psi_r(z)$, is also a function of the distance to the target, but it is dependent on the position of the obstacles, which are assumed to be stationary.

The repulsive potential has some very important properties that need to be examined in order to understand the Secant approach. By examining the repulsive potential function, several limitations can immediately be determined:

- An obstacle at the target location is unresolvable
- A robot in the same position as the obstacle results in infinite potential ($z = o_i$)
- A robot in line with the obstacle and target results in infinite potential ($\theta = \frac{1}{2}\pi$)

These are the immediate limitations that must be placed when examining the field, and they will help to inform the constraints on the initial conditions of the control law. However, a few properties of the Secant approach are also immediately apparent due to the repulsive potential. Including the distance to the target in the numerator of the potential ensures that

the influence of the obstacles will diminish to nothing at the target location. Having the distance to the obstacle in the denominator of the potential ensures that the repulsive force will grow rapidly as the robot approaches the obstacle. Finally, including the secant term in the numerator acts as a penalty for robots behind an obstacle, influencing the robot to avoid and move around to the obstacle towards the target location. These elements to the Secant potential function ensure that the system will converge on the target location.

2.3 Gradient

The gradient will be used in the control law of the Secant approach and is therefore crucial to understand. In order for the origin (the target location) to be a global minimum and to exclude other local minimum, certain properties of the gradient must be examined. The potential field gradient is shown in Equation 2.3. The accompanying variable definitions are shown in Equation 2.4.

$$\frac{\partial \Psi}{\partial z} = k_p z^T + \sum_i \frac{k_i \sec^2 \theta_i}{\|s_i\|^2} z^T \left(\frac{\sec^2 \theta_i}{4 \|s_i\|} z u_i^T P_i + Q_i \right) \tag{2.3}$$

$$u_i = \frac{o_i}{\|o_i\|} \tag{2.4a}$$

$$P_i = I - \frac{s_i s_i^T}{\|s_i\|^2} \tag{2.4b}$$

$$Q_i = I - \frac{z s_i^T}{\|s_i\|^2} \tag{2.4c}$$

The gradient in Equation 2.3 will be used in the control law to determine the motion of the robot. However, since the local minimum issue is by far the greatest concern with APF methods, the property of a single minimum in the field at the target location will also be examined in greater detail before the control law for determining the trajectory within the field is developed. The derivation for the gradient is detailed in Section 2.6.

2.4 Global Minimum

The desire for the Secant approach is to have a single minimum at the target location, which will be a global minimum for the potential field. Showing that the target is a global minimum is quite simple: evaluate the system at $z = 0$ and $z > 0$ as shown in Equation 2.5.

$$\Psi(z = 0) = 0 \tag{2.5a}$$

$$\Psi(z \neq 0) > 0 \tag{2.5b}$$

Since the origin has zero potential, and all other locations must have a greater than zero potential, the origin must be a global minimum of the potential field. Furthermore, the origin can be shown to be the only existing stationary point by examining the potential derivative. Manipulating the gradient in Equation 2.3 yields the equality shown in Equation 2.6.

$$\frac{\partial \Psi}{\partial z} z = \|z\|^2 \left(k_p + \sum_i \frac{k_i \|o_i\|}{\|s_i\|^3} \right) \tag{2.6}$$

Equation 2.6 demonstrates crucial behavior of the gradient within the field. The gradient of the potential function must be bound by the relationship demonstrated in Equation 2.7.

$$\left\| \frac{\partial \Psi}{\partial z} \right\| > k_p \|z\| \tag{2.7}$$

Furthermore, since $k_p \|z\| \geq 0, \forall z \neq 0$, then any location other than the origin must have a non-zero gradient. If no other possible location exists for zero gradient other than the origin, than the origin must be the global minimum of the field.

2.5 Trajectory Planning and Lyapunov

The typical approach to APF path planning is to use the gradient of the potential field as an input to the acceleration of the system. Mass can be neglected by incorporating it with the gain functions, resulting in the system dynamics shown in Equation 2.8, where u is the control input to the system.

$$\ddot{z} = u \tag{2.8}$$

For this analysis, an error velocity will be defined, shown in Equation 2.9. In these equations, v_r is a reference velocity defined by the positive gain constant λ and the distance to the target. The error velocity is therefore the difference between the velocity of the system and the reference velocity.

$$v_r = -\lambda z \tag{2.9a}$$

$$v_e = \dot{z} - v_r \tag{2.9b}$$

$$v_e = \dot{z} + \lambda z \tag{2.9c}$$

Now, a control law can be defined which is informed by the potential function, shown in Equation 2.10.

$$u = \dot{v}_r - k_v v_e - \left(\frac{\partial \Psi}{\partial z}\right)^T \tag{2.10}$$

Using the control law in Equation 2.10 and the system dynamics in Equation 2.8, the state equations are therefore derived in Equation 2.11.

$$\dot{z} = -\lambda z + v_e \tag{2.11a}$$

$$\dot{v}_e = -k_v v_e - \left(\frac{\partial \Psi}{\partial z}\right)^T \tag{2.11b}$$

The state equations are meant to demonstrate that the system will converge on the target location regardless of the layout of the obstacles in the field. To show that this is true, an analysis on the system dynamics using the Lyapunov theorem will be completed in conjunction with the Invariant Set theorem.

The Lyapunov theorem defined in Theorem 3.3 of [47] states: *Assume that there exists a scalar function V of the state x, with continuous first order derivatives such that: $V(x)$ is positive definite; $\dot{V}(x)$ is negative definite; and as $V(x)$ tends towards infinite, $\|x\|$ tends towards infinity as well, then the equilibrium at the origin is globally asymptotically stable.*

As will be shown, this definition will not be sufficient and the Local Invariant Set theorem defined by Theorem 3.4 in [47] will be drawn upon. This theorem states: *Consider an autonomous system of the form $\dot{x} = f(x)$, with f continuous, and let $V(x)$ be a scalar function with continuous first partial derivatives. Assume that: for some $\ell > 0$, the region Ω_ℓ defined by $V(x) < \ell$ is bounded and $\dot{V}(x) \leq 0$ for all x in Ω_ℓ. Let **R** be the set of all points within Ω_ℓ where $\dot{V}(x) = 0$, and **M** be the largest invariant set in **R**. Then, every solution $x(t)$ originating in Ω_ℓ tends to **M** as $t \to \inf$.*

This analysis will show that the target location is globally asymptotically stable for a defined domain within the field while remaining a finite distance away from the obstacles at all times. However, it has previously been shown that a subset of the space defined by obstacles has an infinite potential, and therefore they must be excluded from this analysis. For this reason, the set Ω_ℓ will be defined which is compact (closed and bounded), such that $\Omega_\ell = \{(z, v) \in \mathbb{R}^{2n} : V(z, v) \leq \ell\}$, where ℓ is a positive constant. The set of obstacles and conditions for which $\theta_i = \frac{1}{2}\pi$ will be defined as $\mathcal{O} := \bigcup_i \{z \in \mathbb{R}^n : z = \alpha o_i, \alpha \geq 1\}$.

Since, the set Ω_ℓ is closed and bounded, it naturally excludes the set $\mathcal{O}$ and all of the infinite potentials in the field. Along this set, the positive definite Lyapunov equation is defined in Equation 2.12.

$$V(z, v_e) = \Psi(z) + \frac{1}{2} v_e^T v_e \tag{2.12}$$

The derivative with respect to time of Equation 2.12 is therefore represented in Equation 2.13.

$$\frac{dV}{dt} = \frac{\partial \Psi}{\partial z} \dot{z} + v_e^T \dot{v_e} \tag{2.13}$$

Using values from the state values in Equation 2.11, Equation 2.13 can be rewritten as Equation 2.14.

$$\frac{dV}{dt} = \frac{\partial \Psi}{\partial z} \left(-\lambda z + v_e\right) + v_e^T \left(-k_v v_e - \left(\frac{\partial \Psi}{\partial z}\right)^T\right) \tag{2.14}$$

Simplifying Equation 2.14 by combining like terms, and using the relationship defined by Equation 2.6, the derivative of the Lyapunov equation can be shown to be equal to Equation 2.15, which by definition is negative definite.

$$\frac{dV}{dt} = -\lambda \left(k_p + \sum_i \frac{k_i \left\|o_i\right\|}{\left\|s_i\right\|^3}\right) \left\|z\right\|^2 - k_v \left\|v_e\right\|^2 \leq 0 \tag{2.15}$$

Therefore, so long as the initial conditions begin within the invariant set Ω_ℓ, it will remain within that set and converge on the target location. Since the obstacles are not included within this set, the system will not collide with an obstacle. Furthermore, since Ω_ℓ is bound by a finite value, the system must remain a finite distance away from the obstacles at all times.

2.6 Gradient Derivation

The derivative of the potential function shown in Equation 2.3 is critical to the behavior of the system, and thus the derivation of this function will be demonstrated. In these equations, the generalized position vector x will be used. First, define the function $f_i(x)$ as shown in Equation 2.16.

$$f_i(x) = \frac{(x - o_i)^T u_i}{\|x - o_i\|} \tag{2.16}$$

The intuition of Equation 2.16 is that it is equivalent to $f(x) = -\cos(2\theta_i)$. From these equations, the following relationships must be satisfied shown in Equation 2.17.

$$\frac{\partial f_i}{\partial x} = \frac{u_i^T P_i(x)}{\|x - o_i\|} \tag{2.17a}$$

$$\frac{\partial f_i}{\partial x} x = \frac{\|o_i\| \sin^2(2\theta_i)}{\|x - o_i\|} \tag{2.17b}$$

$$\frac{\partial \theta_i}{\partial x} = \frac{u_i^T P_i(x)}{2\|x - o_i\| \sin(2\theta_i)} \tag{2.17c}$$

$$\frac{\partial \theta_i}{\partial x} x = \frac{\|o_i\| \sin(2\theta_i)}{2\|x - o_i\|} \tag{2.17d}$$

In these equations, P_i is the matrix described by Equation 2.18.

$$P_i(x) = I - \frac{(x - o_i)(x - o_i)^T}{\|x - o_i\|^2} \tag{2.18}$$

Furthermore, define the function $g_i(x)$ by Equation 2.19.

$$g_i(x) = \sec^2(\theta_i) \tag{2.19}$$

Then the relationships detailed in Equation 2.20 must be true.

$$\frac{\partial g_i}{\partial x} = \frac{\sec^4(\theta_i)}{2\left\|x - o_i\right\|} u_i^T P_i(x) \tag{2.20a}$$

$$\frac{\partial g_i}{\partial x} x = \frac{2\tan^2(\theta_i)\left\|o_i\right\|}{\left\|x - o_i\right\|} \tag{2.20b}$$

Define the function $h_i(x)$ as seen in Equation 2.21.

$$h_i(x) = \frac{x^T x}{\left\|x - o_i\right\|^2} \tag{2.21}$$

Then Equation 2.22 can be shown to be true as well.

$$\frac{\partial h_i}{\partial x} = \frac{2x^T Q_i(x)}{\left\|x - o_i\right\|^2} \tag{2.22a}$$

$$\frac{\partial h_i}{\partial x} x = \frac{2\left\|o_i\right\|\left\|x\right\|^2 \cos(2\theta_i)}{\left\|x - o_i\right\|^3} \tag{2.22b}$$

Where in 2.22, Q_i is the matrix defined in Equation 2.23.

$$Q_i(x) = I - \frac{x(x - o_i)^T}{\left\|x - o_i\right\|^2} \tag{2.23}$$

Relabeling the potential function from Equation 2.1 as defined in Equation 2.24 yields:

$$\Psi(z) = \frac{k_p}{2} z^T z + \frac{1}{2} \sum_i k_i g_i(z) h_i(z) \tag{2.24}$$

Using the relationships set in this section, the potential gradient defined in Equation 2.3 and the relationship shown in Equation 2.6 can therefore be shown to be true.

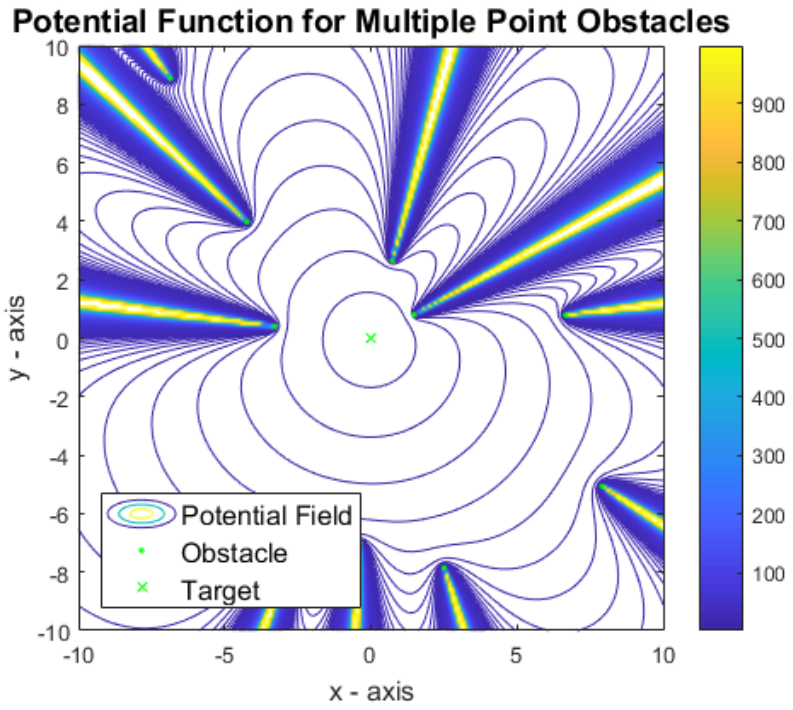


Figure 2.2: A contour plot depicting the values of $\Psi(z)$ for multiple point obstacles randomly placed within the field.

2.7 Results - Potential Field

The analysis so far has demonstrated that the Secant approach only has a global minimum at the target location and is otherwise devoid of local minimum. An example of this in the two dimensional case can be seen in Figure 2.2 with a random configuration of obstacles and the target at the center location. The values of the potential field are not relevant, rather the shape of the field is what matters to the convergence properties of the system.

A closer view of one of the obstacles is shown in Figure 2.3. A very important consideration of the Secant approach is the "comet tail" that follows behind the obstacle relative to the target position. This comet tail is a result of the values for θ_i approaching $\frac{\pi}{2}$. In all dimensions, this comet tail is a line of infinite potential trailing behind the obstacle, but in two dimensions, this feature can make path planning more difficult. The convergence properties of the system still hold, but in practice, the comet tail divides the workspace into viable paths.

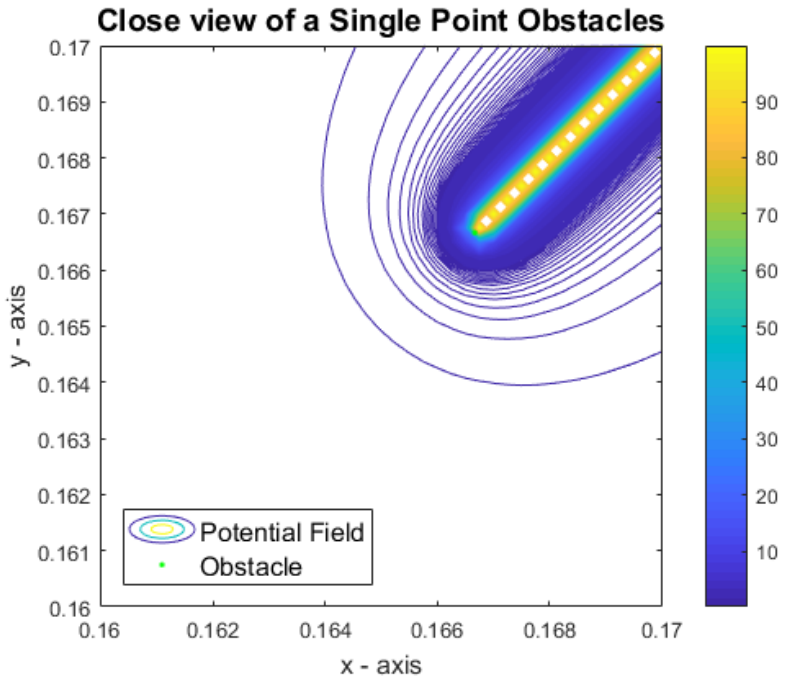


Figure 2.3: Contour plot close up of an obstacle in a potential field. Note, the potential around the obstacle and following the comet tail are so large that the potential not in the immediate vicinity of the obstacle are orders of magnitude lower.

These figures demonstrate the global minimum behavior of the Secant approach which gives the system its desirable behaviors. The system cannot be trapped by a local minimum, which makes this a powerful approach to use with the APF algorithm; however, they still rely on a control scheme to guarantee convergence.

2.8 Results - Control

Demonstrating the convergence of a point robot in an n-dimensional field is a bit challenging. Literally, infinite possible scenarios exist which could be demonstrated, but the system is limited to avoiding points. Therefore, the best way to demonstrate the robustness of the Secant approach in two dimensions is to show it converging in a large field of obstacles. APF methods typically suffer greatly with large numbers of obstacles packed closely together. The Secant approach is demonstrated in Figures 2.4 and 2.5, where a point robot found a trajectory through a field of 196 obstacles to get to a target location. In Figure 2.4

Table 2.1: Gain parameters used for figs 2.4 and 2.5. The system is using generalized units. In theory, the gain parameters do not matter as long as they are positive.

Parameter	**Value**
Attraction Constant (k_p)	500
Repulsion Constant (k_i)	0.001
Damping Constant (k_v)	120

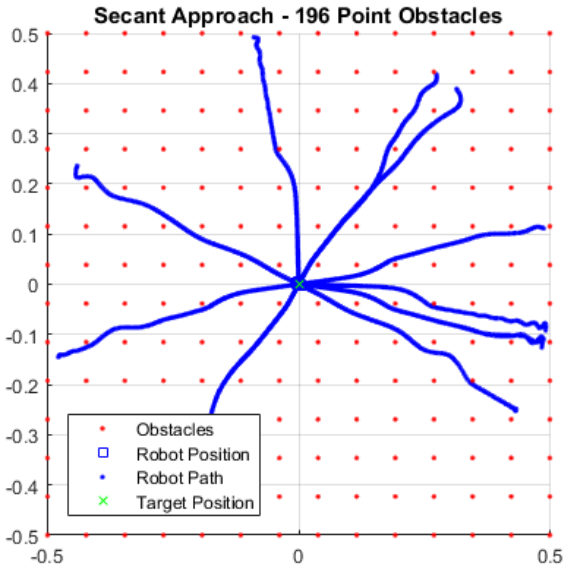


Figure 2.4: The Secant approach pathing through 196 obstacles from a ten random starting positions and converging to [0,0]. At each point in the calculation each of the 196 obstacles were considered for the resulting gradient. In this example, the obstacles are arranged in a grid.

the obstacles are arranged in a grid and in Figure 2.5 the obstacles were distributed randomly.

Figures 2.4 and 2.5 reveal some of the underlying behaviors of the Secant approach. If the initial condition is too close to a Secant line, this might make the numeric calculations challenging to solve while retaining desired accuracy. The oscillation at the beginning of some of the paths are due to a high starting potential, which will be reduced in time with the damping factor. Furthermore, as seen in Table 2.1, the gain on the obstacle function is extremely low compared to the other gains. The reason for this discrepancy is that the repulsive force uses the summation of all obstacles, and thus for large numbers of obstacles

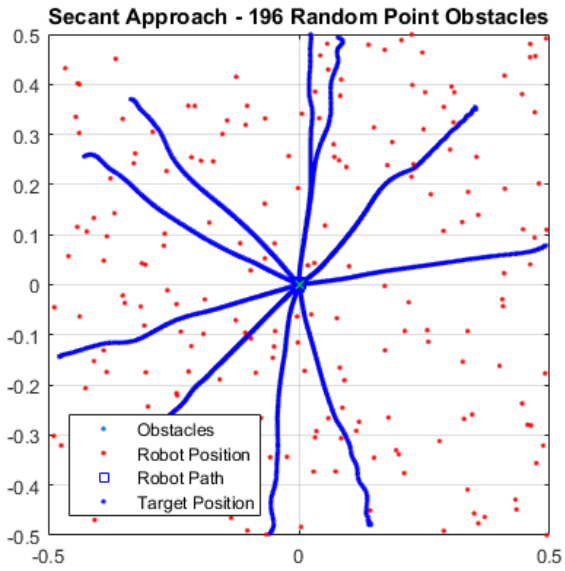


Figure 2.5: The Secant approach pathing through 196 randomly assorted obstacles from ten random starting positions and converging to [0,0]. Just as in Figure 2.4, at each point in the calculation each of the 196 obstacles were considered for the resulting gradient; however, in this example, the obstacle positions were randomly determined.

the gain value can be significantly lowered. Mathematically, the system will converge regardless of the gain parameters; however, in practice this system is solved numerically and may have additional challenges.

2.9 Conclusion

The Secant approach offers a potential function with a guaranteed global minimum at the target location for point robots in an n-dimensional field of point obstacles. By ensuring that the starting position of the robot is not on an existing obstacle or along the comet tail of an obstacle in the field, the control system shown in this chapter will guarantee convergence to the target location. In practice, some consideration must be given to the gain parameters in order to achieve the desired behavior of the system, such as reducing oscillations, but so long as the numerical accuracy is sufficient, the system will converge to the origin of the field while avoiding obstacles. However, although the Secant approach elegantly displays favorable behaviors, it is not unique with its convergent properties; the TAPF approach will

be shown to share this convergent behavior.

CHAPTER 3

THE TRAVELING ARTIFICIAL POTENTIAL FIELD APPROACH

3.1 Introduction

Catering potential fields to have desired convergent properties is challenging, but not impossible. The Secant approach is a way of applying the APF algorithm that is provably globally convergent for point obstacles and robots in an n-dimensional space. However, other APF methods that are globally convergent are also possible, such as the Traveling Artificial Potential Field (TAPF) approach. The gradient for the Secant approach solely considers the relative position of the targets and obstacles, but the TAPF method also considers the velocity of the robot relative to the obstacles (hence: "Traveling" APF). In accounting for the velocity of the robot, the influence of the obstacles will be different based on the robot's trajectory, which may be more desirable than the Secant approach in various applications. Like the Secant approach, the TAPF approach has constraints on the initial condition, but is otherwise guaranteed to reach the target location. The net result is another APF method which is globally convergent regardless of the number of obstacles in the field.

The TAPF approach will be examined in detail in this analysis, but a few aspects of the control method should be noted. A similar requirement on the initial conditions exists for this method as with the Secant approach. The starting position of the point robot cannot be perfectly in line with the obstacle and the target location or on the obstacle itself. Since this is the greatest restriction on the Secant approach, the two methods are therefore nearly identical in when they might be applied. The only significant difference between the approaches is in how the trajectory performs. By considering the velocity of the robot, the force applied has a natural damping factor that potentially leads to smoother paths. Also, the TAPF approach will be proven to be convergent using a very similar method to the

Secant approach, so further additions to the theory will be equally applicable to the Secant and TAPF approaches.

3.2 Traveling Force

The TAPF approach applies a force to the robot that is dependent on the robot's position and velocity. This method is a departure from the Secant approach, which used a potential gradient to dictate the robot's motion. However, the development of this force is based on observations made on a modified Secant potential function. This modification, which applies a normalizing term to the gradient is shown in Equation 3.1, with $\Phi(z)$ as the modified potential function.

$$\Phi(z) = \frac{1}{2}k_p z^T z + \frac{1}{2}\sum_i k_i \ln\left(1 + \frac{z^T z \sec^2(\theta_i)}{\|z - o_i\|^2}\right) \tag{3.1}$$

The derivative of this function is developed in a very similar way to the Secant potential function shown in Section 2.6, with the result depicted in Equation 3.2. All relevant relationships and definitions for this section are listed in Equation 3.7.

$$\frac{\partial \Phi}{\partial z} = k_p z^T + \sum_i k_i \alpha_i(z) z^T \left(\frac{\sec^2(\theta_i)}{4\|z - o_i\|} z u_i^T P_i + Q_i\right) \tag{3.2}$$

From Equation 3.2, an important relationship is discerned regarding the shape of the slope, shown in Equation 3.3. This relationship will help to show the convergence properties of the system's control scheme.

$$\frac{\partial \Phi}{\partial z} z = \left[k_p + \sum_i \frac{k_i \beta_i(z) \|o_i\|}{\|z - o_i\|}\right] \|z\|^2 \geq k_p \|z\|^2 \tag{3.3}$$

Simplifying and rearranging the potential due to obstacle forces, $\frac{\partial \Phi_o}{\partial z}$, yields Equa-

tion 3.4, which will help to produce important insights into the nature of this gradient.

$$\frac{\partial \Phi_o}{\partial z} = \frac{k_i}{\|s_i\|}\left(\frac{\gamma_i(z)}{4}\sec^2(\theta_i)P_i u_i - \alpha_i(z) z^T o_i \frac{s_i}{\|s_i\|}\right) \tag{3.4}$$

Equation 3.4 can be shown to be comprised of two components: a normal force, F_r, and a tangential force, F_t, depicted in Equation 3.5.

$$F_r = z^T o_i \frac{s_i}{\|s_i\|^2} \tag{3.5a}$$

$$F_t = -\frac{\tan(\theta_i)}{2\|s_i\|}\frac{\bar{s}_i}{\|\bar{s}_i\|} \tag{3.5b}$$

The forces shown in Equation 3.5 can now be used as a basis for the TAPF approach. The force applied to the system, defined by F_T, is comprised of an attractive force and repellent forces, shown in Equation 3.6. In this equation, $\mu(x)$ is a one sided ramp, such that $\mu(x) = 0, x \leq 0$ and $\mu(x) = x, x > 0$. The remaining variables are listed in Equation 3.7.

$$F_T = k_p z + \sum_i \tilde{k}_i \frac{\mu\left(\bar{s}_i^T v t_i\right) t_i}{4\|s_i\|^2 \|\bar{s}_i\|^4}\bar{s}_i + \bar{k}_i \frac{s_i^T v}{\|s_i\|^4} s_i \tag{3.6}$$

Principally, the force used in the TAPF approach are derived from the modified gradient function, but they are not in themselves a gradient. Also note that α_i and γ_i in Equation 3.7 are bounded functions; they are used for a normalization factor.

$$s_i = z - o_i \tag{3.7a}$$

$$u_i = \frac{o_i}{\|o_i\|} \tag{3.7b}$$

$$\bar{s}_i = P_i u_i \tag{3.7c}$$

$$t_i = \tan(\theta_i) \tag{3.7d}$$

$$\bar{k}_i = k_i \|o_i\| \alpha_i(z) \tag{3.7e}$$

$$\tilde{k}_i = k_i \gamma_i(z) \tag{3.7f}$$

$$P_i(z) = I - \frac{(z - o_i)(z - o_i)^T}{\|z - o_i\|^2} \tag{3.7g}$$

$$\alpha_i(z) = \left(\|z - o_i\|^2 \cos^2(\theta_i) + \|z\|^2\right)^{-1} \tag{3.7h}$$

$$\gamma_i(z) = \frac{(\|z - o_i\| + \|o_i\|)^2}{\|z - o_i\|^2 \cos^2(\theta_i) + \|z\|^2} \tag{3.7i}$$

$$\beta_i(z) = \frac{\cos^2(\theta_i)}{\|z - o_i\|^2 \cos^2(\theta_i) + \|z\|^2} \tag{3.7j}$$

3.3 Control

The reason for the complexity of the TAPF force is to ensure that the system is globally asymptotically stable at the origin, as will be seen in this analysis. The control law, seen in 3.8, is identical to the control law seen for the Secant approach seen in Equation 2.10, using the same state equations seen in Equation 2.11 as well.

$$u = \dot{v}_r - k_v v - F_T \tag{3.8}$$

From the control law, the closed loop dynamics can be seen in Equation 3.9.

$$\dot{z} = -\lambda z + v \tag{3.9a}$$

$$\dot{v} = -k_p z - k_v v - F_T \tag{3.9b}$$

In order to satisfy the Lyapunov equation, the definitions of the domain of the starting conditions is the same as listed in Section 2.5. This implies that the invariant set Ω_ℓ, which includes the origin and the initial condition does not contain the obstacles or their comet tails. With this restriction, the Lyapunov equation is a summation of the potential and kinetic energy of the system, shown in Equation 3.10.

$$V(z, v) = \Phi(z) + \frac{1}{2} v^T v \tag{3.10}$$

The derivative of the Lyapunov equation is simply shown in Equation 3.11.

$$\frac{dV}{dt} = \frac{\partial \Phi}{\partial z} \dot{z} + v^T \dot{v} \tag{3.11}$$

Expanded, the Lyapunov function is seen in Equation 3.12.

$$\frac{dV}{dt} = \frac{\partial \Phi}{\partial z} \left(-\lambda z + v\right) - v^T \left(\sum_i \tilde{k}_i \frac{\mu \left(\bar{s}_i^T v t_i\right) t_i}{4 \left\|s_i\right\|^2 \left\|\bar{s}_i\right\|^4} \bar{s}_i + \bar{k}_i \frac{s_i^T v}{\left\|s_i\right\|^4} s_i + k_v v \right) \tag{3.12}$$

Furthermore, the derivative of the Lyapunov equation can be shown to be less than Equation 3.13.

$$\begin{aligned} \frac{dV}{dt} \leq & -k_v \left\|v\right\|^2 - \lambda k_p \left\|z\right\|^2 \\ & + \sum_i \frac{k_i}{\left\|s_i\right\|} \left(\frac{\gamma_i}{2} \frac{\bar{s}_i^T v t_i}{\left\|\bar{s}_i\right\|} - \alpha z^T o_i \frac{s_i^T v}{\left\|s_i\right\|} \right) - \bar{k}_i \frac{\left(s_i^T v\right)^2}{\left\|s_i\right\|^4} - \tilde{k}_i \frac{\mu \left(\bar{s}_i^T v t_i\right) \bar{s}_i^T v t_i}{4 \left\|s_i\right\|^2 \left\|\bar{s}_i\right\|^4} \end{aligned} \tag{3.13}$$

Equation 3.13 reduces to Equation 3.14

$$\frac{dV}{dt} \leq -k_v \|v\|^2 - \lambda k_p \|z\|^2 + \sum_i \bar{k}_i \frac{\|z\|}{\|s_i\|^2} |s_i^T v| - \bar{k}_i \frac{\left(s_i^T v\right)^2}{\|s_i\|^4} + \tilde{k}_i \frac{\mu(\bar{s}_i^T v t_i)}{2 \|s_i\| \|\bar{s}_i\|} - \tilde{k}_i \frac{\mu(\bar{s}_i^T v t_i)^2}{4 \|s_i\|^2 \|\bar{s}_i\|^4} \tag{3.14}$$

Using a completing the square technique, Equation 3.14 is equal to Equation 3.15.

$$\frac{dV}{dt} \leq -k_v \|v\|^2 - \left(\lambda k_p - \frac{1}{4} \sum_i \bar{k}_i + \frac{\tilde{k}_i}{\|o_i\|^2}\right) \|z\|^2 - \sum_i \bar{k}_i \left(\frac{1}{2} \|z\| - \frac{|s_i^T v|}{\|s_i\|^2}\right)^2 - \tilde{k}_i \left(\frac{1}{2} \|\bar{s}_i\| - \frac{\mu\left(\bar{s}_i^T v t_i\right)}{2 \|s_i\| \|\bar{s}_i\|^2}\right)^2 \tag{3.15}$$

Crucially, Equation 3.15 has a term which is not guaranteed to be negative. This discrepancy can be resolved by choosing gain values which satisfy Equation 3.16.

$$\tilde{k}_p = \lambda k_p - \frac{1}{4} \left(\sum_i \bar{k}_i + \frac{\tilde{k}_i}{\|o_i\|^2}\right) > 0 \tag{3.16}$$

With Equation 3.16, the derivative of the Lyapunov equation is shown to be bounded by a system which is negative definite, shown Equation 3.17, guaranteeing that the origin is globally asymptotically stable.

$$\frac{dV}{dt} \leq -\tilde{k}_p \|z\|^2 - k_v \|v\|^2 \leq 0 \tag{3.17}$$

3.4 Results

The results of the TAPF approach will closely mirror the results of the Secant approach demonstrated in Chapter 2. Figures 3.1 and 3.2 show the TAPF approach solving the same field of obstacles that was demonstrated in Figures 2.4 and 2.5 and using the same gain

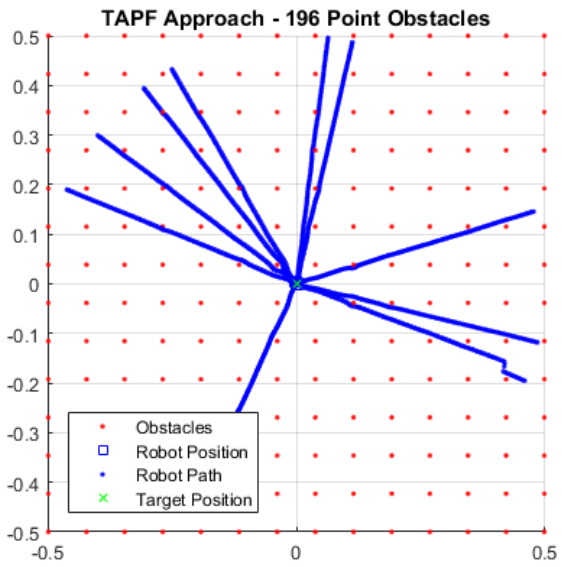


Figure 3.1: The TAPF approach pathing through 196 obstacles from ten random starting positions and converging to [0,0]. At each point in the calculation each of the 196 obstacles were considered for the resulting force, the Secant approach equivalent is shown in Figure 2.4.

parameters listed in Table 2.1. As expected, the system converges to the target location despite the large number of obstacles in the field.

Comparing the Secant and TAPF approaches in Figures 2.4 and 2.5 and Figures 3.1 and 3.2 respectively shows some interesting differences between the two methods. The TAPF approach has much less oscillation than the Secant approach and appears to get closer to the obstacles while still avoiding them. This behavior is due to the consideration of the velocity within the applied force, which acts as a dampener in the state equations. Both systems are still perfectly valid, but different approaches can be chosen based on the desired trajectory of the system while it converges to the target.

3.5 Conclusion

The most significant result of this analysis is that a second APF algorithm exists which can be proven to be globally asymptotically stable at the target location. The TAPF approach does not significantly deviate from the Secant method in its use or performance,

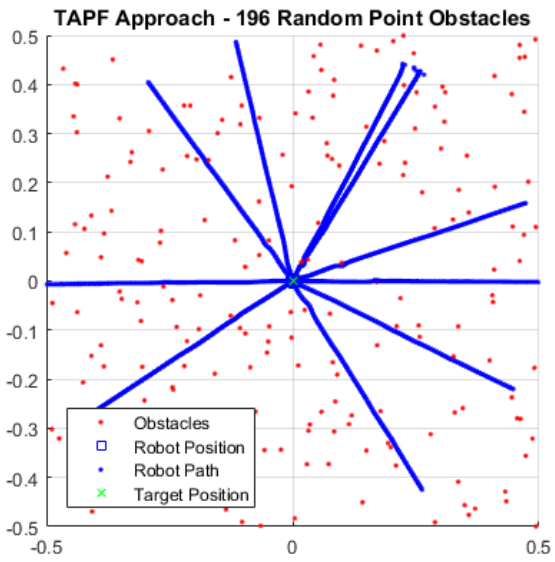


Figure 3.2: The TAPF approach pathing through 196 randomly assorted obstacles from ten random starting positions and converging to [0,0]. At each point in the calculation each of the 196 obstacles were considered for the resulting force, the Secant approach equivalent is shown in Figure 2.5.

but it does demonstrate that various techniques can be applied for desired trajectory behavior. The TAPF force results in a generally smoother curve than the Secant approach, since it accounts for the relative velocity of the robot with respect to obstacles. However, the formulation of the TAPF force is more difficult, since limitations are imposed on the gain parameters. Overall, the TAPF approach is another APF method which can be used to globally converge on a target using a point robot in a field of point obstacles in n-dimensional space.

CHAPTER 4
THE BOUNDARY LAYER

4.1 Introduction

The Secant and TAPF approaches have been shown to be globally convergent on their target locations for point obstacles in n-dimensional space. However, treating obstacles as points is a sever limitation when trying to apply the theory to real world applications. In physical circumstances, obstacles will have solid dimensions. APF methods do not directly approach this problem. Typically, the potential field is altered to account for the physical size of the obstacle. However, the Secant and TAPF approaches have very well defined properties that should be conserved if possible. Therefore, a new way of accounting for obstacles has been developed: the boundary layer method. If a transformation were possible that could treat a solid bodied obstacle as if it were a point, the convergence properties of the Secant and TAPF methods would be preserved. This transformation is what the boundary layer attempts to accomplish. When the robot is outside of a boundary layer, the obstacle appears as a point. Inside the boundary layer, the space is "squeezed" between the radius of the obstacle and the boundary layer, such that the distance between the boundary layer and the radius appears mathematically the same as the distance between the boundary layer and the center of the obstacle. This spatial transform ensures that the mathematical properties of the Secant and TAPF approach are preserved, but it also imposes new restrictions on the field.

The intention of this analysis is to demonstrate the simplest practical application of the Secant and TAPF approaches. This analysis still considers the robot to be a point in n-dimensional space but now allows for non-point obstacles to be examined as well. Since the boundary layer method relies on a transformation of the APF potential functions, the

restrictions on the original equations will cary through. Furthermore, most of this analysis will be demonstrated using the Secant approach, but the methodology is equally valid for the TAPF potential function as well. The Secant and TAPF approaches have the same conditions for their convergence, so their application in the boundary layer method is the same. Overall, the use of a boundary layer will allow for the expansion of obstacles from points to solids, making the system far more applicable.

The boundary layer method applies a smooth, non-linear transformation within a bounded region of space surrounding an obstacle. An example of this is shown in Figure 4.1. In this example, a point robot that has entered into the boundary layer will mathematically appear as if it is closer to the obstacle than it actually is. At the radius of the obstacle, it will mathematically appear as if it were at the center of the obstacle for the purposes of control. The details of the transformation functions will be explained within this analysis, but this transformation allows for the potential function generated by point obstacles to be applied to solid obstacles.

4.2 The Boundary Layer Transform

To begin this analysis, certain terms must be well defined in order to understand their behavior. First, let the i'th obstacle be represented by $\mathcal{O}_i$, which can be any compact set with a well defined bounding radius. Let the obstacle and the boundary layer be defined by Equation 4.1.

$$\mathcal{O}_i = \left\{ z : \|z - o_i\| \leq \rho_i \left(\frac{z - o_i}{\|z - o_i\|} \right) \right\} \tag{4.1}$$

Furthermore, define the boundary layer of the obstacle such that: $\mathcal{B}_i = \mathcal{O}_i + \bar{B}_{\delta_i}$, where $\bar{B}_{\delta_i} > 0$ is a closed ball, which assumes that the boundary layer has a non-zero radius. Furthermore, this analysis require the restriction that $\mathcal{B}_i \cap \mathcal{B}_j = \varnothing, \forall i, j$ and $0 \notin \mathcal{B}_i$, which precludes boundary obstacle layers from overlapping and the origin from being within a

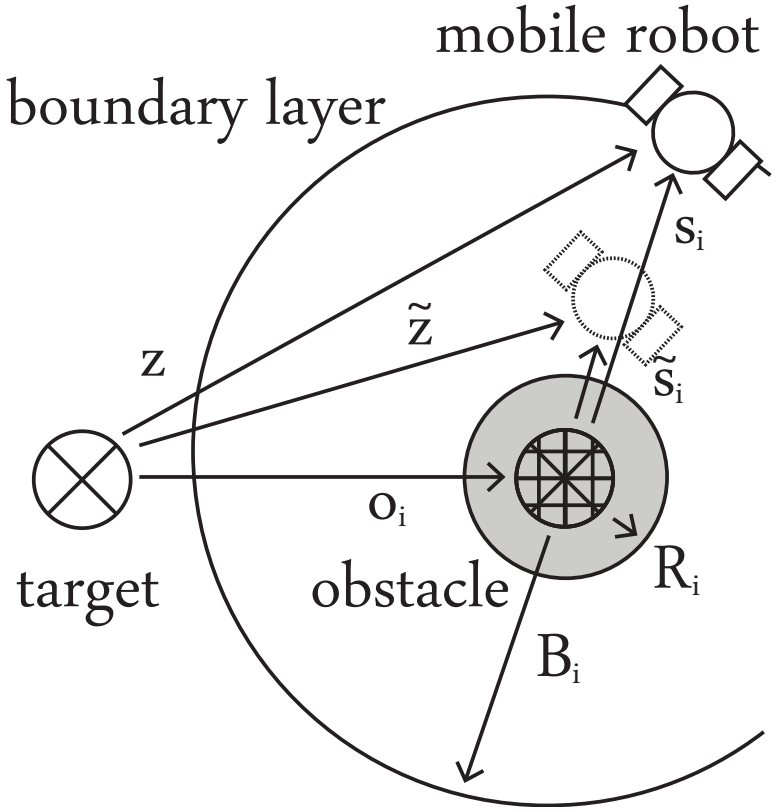


Figure 4.1: An example of a robot in the proximity of a target and an obstacle with a boundary layer. This is a modified version of the diagram in Figure 2.1, such that now the obstacle is a solid. In this figure, R_i and B_i are scalar radius values of the obstacle and boundary layer; $\tilde{z}$ and $\tilde{s}$ are modified distance values to the robot within the boundary layer.

boundary layer. From this, the boundary layer transform can now be defined, as shown in Equation 4.2, with $w_i = \frac{s_i}{\|s_i\|}$.

$$\phi(z) = \begin{cases} z - \rho_i(w_i)\sigma(\xi_i)w_i, \ z \in \mathcal{B}_i \\ z, \ z \notin \cup_i \mathcal{B}_i \end{cases} \tag{4.2}$$

The physical intuition of Equation 4.2 is that outside of an obstacle's boundary layer, the space is untransformed. Inside of the boundary layer, the transform shifts the apparent position of the robot closer to the center of the obstacle. The transformation is reliant on the radius of the obstacle, ρ_i, and the nonlinear transform $\sigma(\xi_i)$. In this equation, $\sigma(\xi_i)$ is a monotonically increasing function between zero and one, and ξ_i describes the space between the boundary layer and the radius of the obstacle, shown in Equation 4.3.

$$\xi_i = \frac{1}{\delta_i}\left(\rho_i(w_i) + \delta_i - \|s_i\|\right) \tag{4.3}$$

A simple equation for $\sigma(x)$ is the parabolic function $\sigma(x) = x^2$. This equation satisfies a zero initial slope and varies from 0 to 1 at $x = 1$ continuously. The functions σ, ξ_i, and $\psi(z)$ defines how the robot will behave in the boundary layer. At the boundary layer, $\|s_i\| = \rho_i(v_i) + \delta_i$, and thus $\xi_i = 0$ and $\phi(z) = z$. This function is continuous between the inside and outside of the boundary layer. However, to understand the relationship of the boundary layer in completion, the inverse of the transform must also be understood, as seen in Equation 4.4. In this equation, $\tilde{z} = \phi(z)$, and $\tilde{s}_i = \phi(z) - o_i$.

$$\phi^{-1}(\tilde{z}) = \begin{cases} \tilde{z} + \rho_i(v_i)\sigma(\xi_i)v_i, \ \tilde{z} \in \mathcal{B}_i \\ \tilde{z}, \ \tilde{z} \notin \cup_i \mathcal{B}_i \end{cases} \tag{4.4}$$

where $\xi_i = \frac{\sqrt{(2\rho_i(w_i)+\delta_i)^2 - 4r_i\|\tilde{s}_i\|} - \delta_i}{2r_i}$, $\tilde{s}_i = \|\phi(z) - o_i\|$, and $w_i = \frac{\tilde{s}_i}{\|\tilde{s}_i\|}$. This also satisfies the inequality $\frac{\|\phi(z)-o_i\|}{\|z-o_i\|-\rho_i(v_i)} \leq \frac{2\rho_i(w_i)+\delta_i}{\delta_i}$, $\forall z \in \mathcal{B}_i$. Furthermore, both ϕ and ϕ^{-1} are C^1 functions with their Jacobians $\partial\phi/\partial z$ and $(\partial\phi/\partial z)^{-1}$ are well defined at every $z \notin \mathcal{O}_i$ and

$\tilde{z} \neq o_i$, respectively.

The boundary layer method perserves the convergence properties of the Secant and TAPF approaches previously defined. The boundary layer is guaranteed to not create local minimums and it does not affect the stability properties of the Lyapunov equations defined in Eqs. 2.12 and 3.10. To show that this is true, the system can be simplified to scenarios inside and outside of the boundary layer. Any location outside of the boundary layer is defined by z and has been shown to be convergent on the target. Inside the boundary layer, the position is defined as $\tilde{z} = \phi(z)$. The analysis has already shown that the global minimum has already been proven to be true for $\Psi(z)$; thus to satisfy the boundary layer condition it must also be shown to be true for $\tilde{\Psi}(z) := \Psi \circ \phi(z) = \Psi(\tilde{z})$. This is exemplified in the chain rule showed in Equation 4.5.

$$\frac{\partial \tilde{\Psi}}{\partial z} = \frac{\partial \tilde{\Psi}}{\partial \tilde{z}} \frac{\partial \psi}{\partial z} \neq 0 \tag{4.5}$$

Equation 4.5 is true everywhere except for at the origin, $z = 0$, and at the center of an obstacle $\tilde{z} = o_i$. Since the same is true for $\frac{\partial \tilde{\Psi}}{\partial \tilde{z}}$ and the invertibility of $\frac{\partial \psi}{\partial z}$ as described by Equation 4.4, then it must be true that the origin is a local minimum for $\tilde{\Psi}(z)$ and this is the only minimum in the field. Thus, the origin is the global minimum for both $\Psi(z)$ and $\Psi(\tilde{z})$.

4.3 Lyapunov Stability with Solid Obstacles

Just as the global minimum needed to be shown to be consistent with $\tilde{\Psi}(z)$, the control law needs to be redefined for within the boundary layer and shown to be asymptotically stable. For this transformation, the state vector will be redefined to the variables $\tilde{z}$ and $\tilde{v}_e$, where $\tilde{v}_e = \frac{\partial \phi}{\partial z}\dot{z} + \lambda \tilde{z}$. The control law is then defined as shown in Equation 4.6.

$$u = \left(\frac{\partial \phi}{\partial z}\right)^{-1} \left(-k_v \tilde{v}_e - \kappa(z, \dot{z}) - \left(\frac{\partial \Psi(\tilde{z})}{\partial \tilde{z}}\right)^T\right) \tag{4.6}$$

In this equation, it is necessary to define $\kappa(z, \dot{z})$ such that $\kappa(z, \dot{z}) = \frac{d}{dt}\left(\frac{\partial \phi}{\partial z}\right)\dot{z} + \lambda \frac{\partial \phi}{\partial z}\dot{z}$. From the control input, the state equations for $\tilde{z}$ and $\tilde{v}_e$ are defined in Eqs. 4.7a and 4.7b.

$$\dot{\tilde{z}} = -\lambda \tilde{z} + \tilde{v}_e \tag{4.7a}$$

$$\dot{\tilde{v}}_e = -k_v \tilde{v}_e - \left(\frac{\partial \Psi}{\partial \tilde{z}}\right)^T \tag{4.7b}$$

From the state equations and using the same logic depicted in Section 2.5, the system can clearly be shown to be globally asymptotically stable. Given the invertbility properties of ϕ, the closed–loop system has guaranteed collision avoidance with solid obstacles as long as the initial position of the robot does not begin inside of the obstacle. This can be demonstrated using the Lyapunov function shown in Equation 4.8.

$$V(\tilde{z}, \tilde{v_e}) = \Psi(\tilde{z}) + \frac{1}{2}\tilde{v}_e^T \tilde{v}_e \tag{4.8}$$

Using parallel methods to those described in Sections 2.5 and 3.3 for the Secant and TAPF approach respectively demonstrate the asymptotic stability of the system, despite the additional constraints of the boundary layer.

4.4 Results: Potential Field

A crucially important aspect of the APF algorithm in combination with the boundary layer will now be explored. In the analysis of the boundary layer, it is assumed that the radius is continuous and that there is only one mapping of the radius for any angle of the obstacle. This implies that the radius does not need to be constant. Indeed, a subset of convex shapes also satisfies these conditions. Thus, the boundary layer extends the Secant and TAPF approaches to real-world environments, where the shape of obstacles cannot always assume to have a constant radius. Figure 4.2 demonstrates a field of circular obstacles randomly arranged with the potential function from the Secant approach. Figures 4.3 and

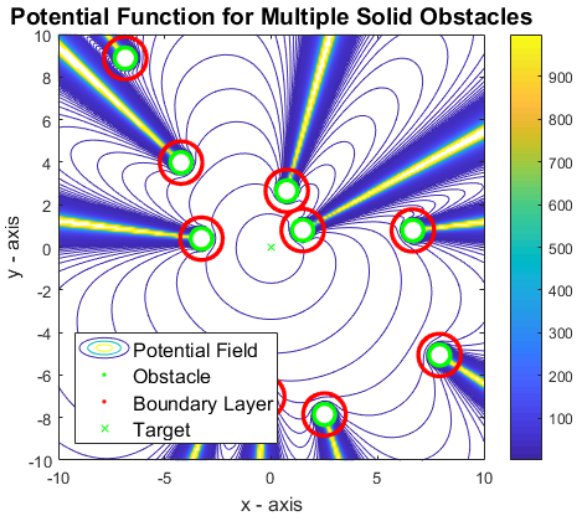


Figure 4.2: A contour plot depicting potential for the Secant approach for multiple solid, circular obstacles randomly placed within the field.

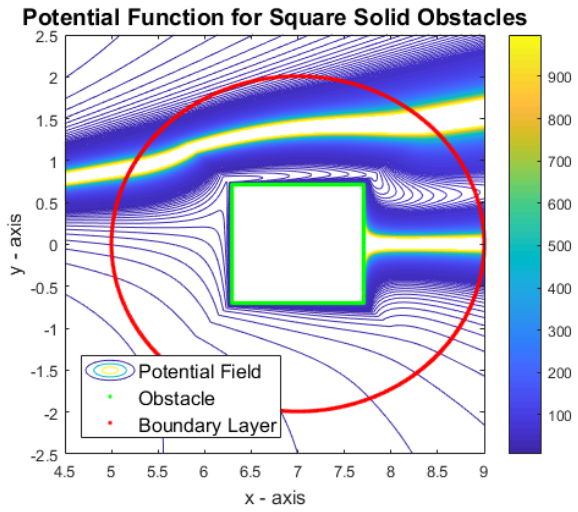


Figure 4.3: A close up depiction of a field of obstacles shaped as squares using the Secant approach. In this image, the channel between the potential functions of multiple obstacles can be seen despite the varying radius of the obstacle.

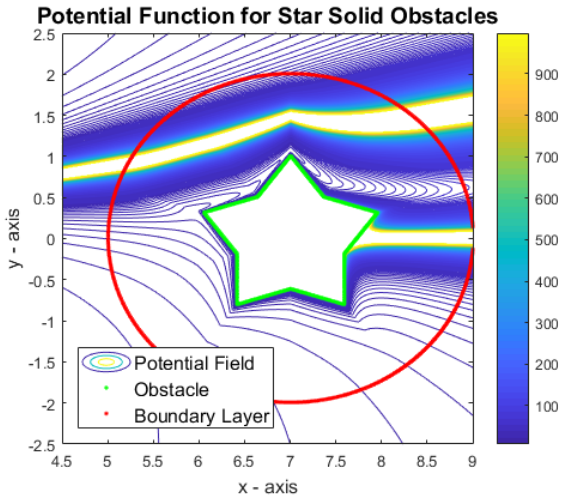


Figure 4.4: A close up depiction of a field of obstacles shaped as stars using the Secant approach. This image is noticeably convex, and yet local minimums do not arise.

4.4 are examples of the Secant approach potential demonstrating that any continuous shape that can be described in polar coordinates will not create a global minimum.

4.5 Results: Control

As interesting as the potential function is for various non-uniform objects, the crucial application of the boundary layer is in the control for solid bodied obstacles. To demonstrate the behavior of the boundary layer for various obstacles, single obstacle fields were contrived to demonstrate convergence for both the Secant and TAPF approaches. In Figures 4.5 and 4.7, a point robot avoids a square obstacle. Likewise, in Figures 4.7 and 4.8, a point obstacle avoids a star shaped obstacle. From these figures, some of the differences in behaviors between the Secant and TAPF approaches can be visualized. The TAPF approach has natural dampening arising from the consideration of the robot's trajectory in its force function. This causes the TAPF approach to more closely conform to the shape of the obstacle, something which may or may not be desirable in the system. To complete the demonstration of the Secant and TAPF approach for solid obstacles, each method experienced ten random starting locations outside of a field of 196 obstacles and converged on a target, as shown in Figures 4.9 and 4.10.

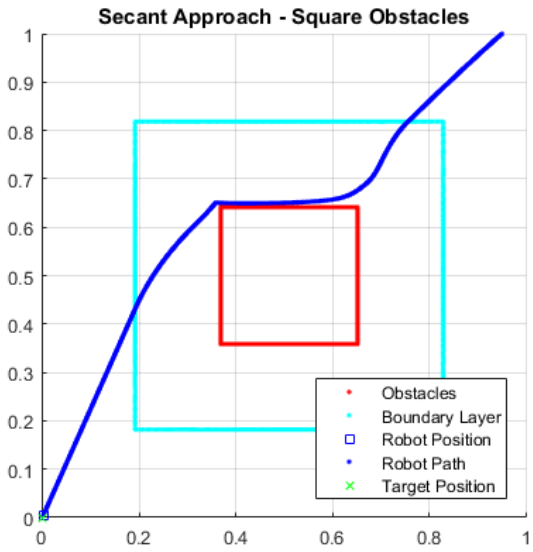


Figure 4.5: The Secant approach applied to a single square obstacle.

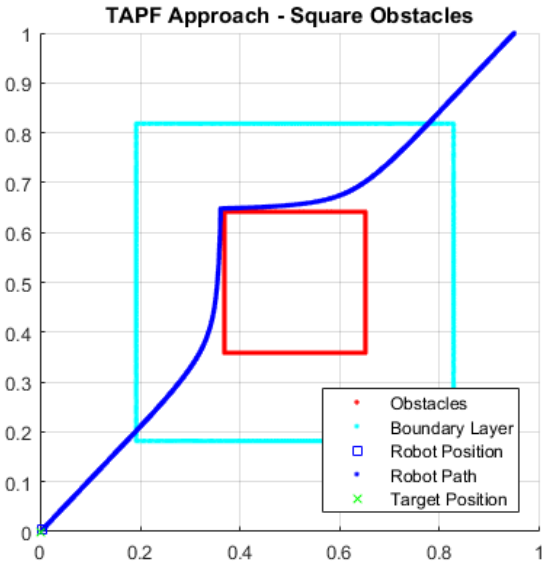


Figure 4.6: The TAPF approach applied to a single square obstacle.

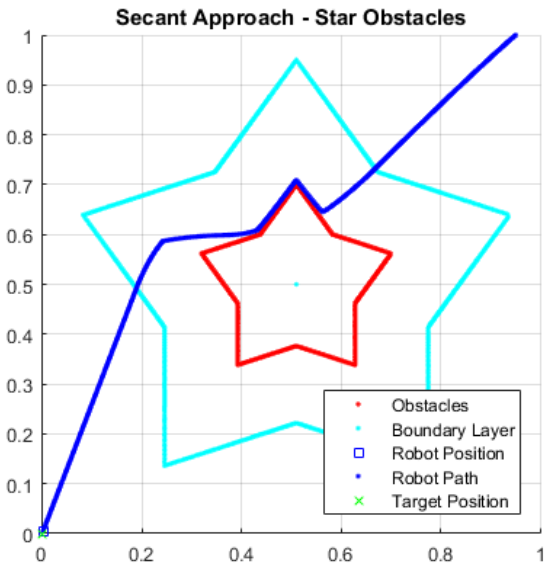


Figure 4.7: The Secant approach applied to a single star obstacle.

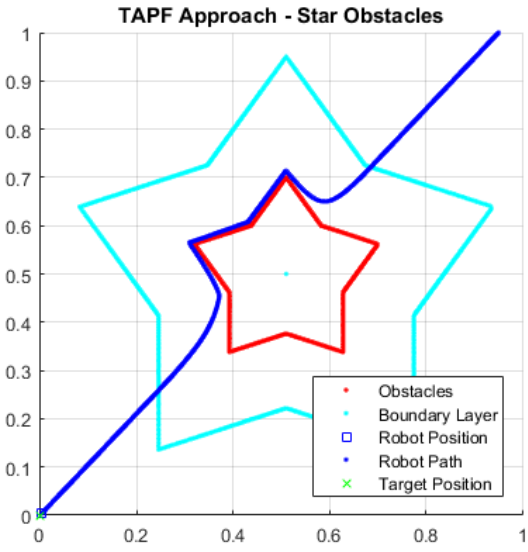


Figure 4.8: The TAPF approach applied to a single square obstacle.

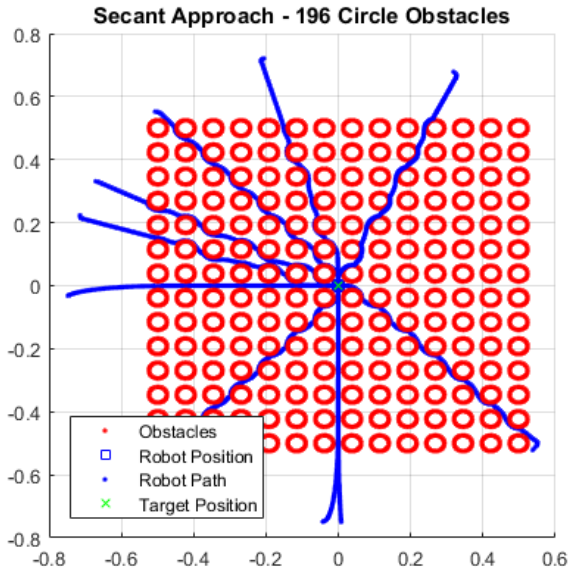


Figure 4.9: The Secant approach pathing through 196 obstacles from a ten random starting positions and converging to [0,0]. At each point in the calculation each of the 196 obstacles were considered for the resulting gradient.

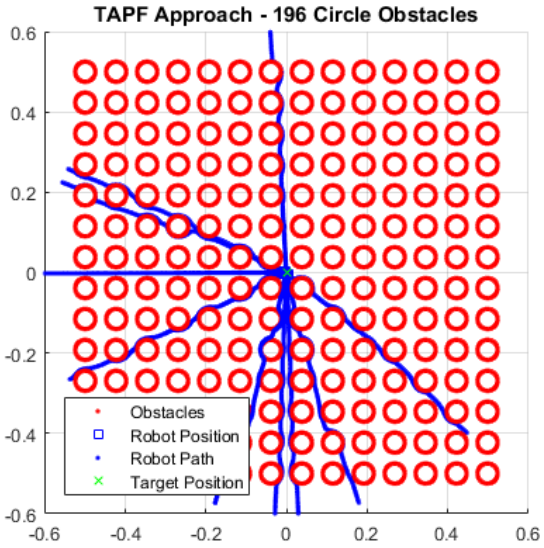


Figure 4.10: The TAPF approach pathing through 196 obstacles from a ten random starting positions and converging to [0,0]. At each point in the calculation each of the 196 obstacles were considered for the resulting force.

Each of these examples demonstrate that the convergence property of the Secant and TAPF approaches can be conserved when applied to solid obstacles. However, these examples have been limited to two dimensions for demonstration purposes. Crucially, the boundary layer transformation is applicable in n-dimensional space.

4.6 Conclusion

The Secant and TAPF approaches have previously been shown to be valid trajectory planning methods for point robots navigating through a field of point obstacles in n-dimensional space. However, real-world examples require the ability to consider obstacles of finite size in order to avoid intersections reliably. The boundary layer method transforms the space around obstacles such that they appear to be points in space, and the application of this transformation allows for robots to avoid the solid obstacles as if they were points. The execution of the boundary layer results in an apparent shifting of the robot position in relationship to the target and the obstacles. This shift artificially increases the force experienced by the robot in proportion to its distance to the bounds of the obstacle. This shifting is what allows for the obstacles to be avoided as if they were points: mathematically, the solid obstacles can be treated as if they were points. By transforming solid obstacles in this way, the underlying benefits of the Secant and TAPF approaches are preserved. As long as the system does not have overlapping boundary layers and the target position is not within a boundary layer, the system will converge. Expanding the Secant and TAPF approaches to account for solid bodied obstacles of arbitrary size greatly increases their applicability to real-world scenarios.

CHAPTER 5
HIGHER ORDER SYSTEMS COMPARISON

5.1 Introduction

Although most of the figures and analysis has been for two dimensional cases, the APF scheme lends itself naturally to higher order systems. Indeed, the major advantage of the APF algorithms, and the Secant and Traveling approaches in particular, is that they are generalizable to n-dimensional systems. Furthermore, as will be demonstrated within this chapter, the APF approaches grow linearly with dimension, whereas search methods grow exponentially. To show how well the APF approaches compare to traditional search methods, the A* algorithm will be tested alongside the Secant approach, and the performances of each algorithm will be evaluated. To note, the main criteria that will be evaluated are the number of iterations necessary for convergence and the accuracy of the algorithms. The amount of time that each iteration takes will only be measured for the purpose of showing the rate of growth of the algorithm. Optimization techniques, both in the coding and in the processing, as well as improved hardware, could decrease the computation time for these methods. An important consideration is that a myriad of search methods exist, some of which may converge in fewer iterations with better accuracy than A* in certain situations. This comparison is not meant to conclusively demonstrate when A* is more applicable than the APF methods, nor is the purpose to say that search methods are without merit. Rather, this examination is meant to demonstrate the behavior of search methods as they grow in accuracy and dimension.

This analysis will begin with a very brief look at the A* method, including a discussion of its application and a comparison to the Secant approach. The algorithms will then be implemented in a simple path planning routine in two dimensions. The cell size utilized

in the A* method will then be adjusted to demonstrate how the number of iterations to convergence grows with accuracy. Then, the dimension of the algorithm will be increased to demonstrate how both the Secant approach and A* algorithms grow with dimensions. These growth rates will demonstrate how the APF methods are much more applicable for higher order systems in terms of computational efficiency.

5.2 Algorithm Growth

The A* algorithm will be examined in brief so that the comparison to the APF methods can be properly contextualized. Figure 5.1a shows the basic A* algorithm. The figure is obviously simplified into steps that could be further subdivided, but this shows a general understanding of the process involved. An initial phase is necessary to retract the space into cells or nodes which can then be searched, and then an iterative process explores the space to find a path from the starting position to the goal. The dimensional growth of the system is actually originated from two steps within this process. "Generate map" and "Add neighbors to the open list" are each dependent on the dimension of the system and expand exponentially. "Generate map" can be precomputed and only needs to be done once (so long as the field is static and perfectly known), but "Add neighbors to the open list" has to be done iteratively, and the number of neighbors is exponentially related to the size of the system's dimension. Tactics can be employed to mitigate the number of cells within the map being searched, and the heuristic values can be tuned to improve convergence time, but the nature of the algorithm causes inherent, exponential growth for each additional dimension.

To contrast, Figure 5.1b demonstrates the APF algorithm in similar terms to Figure 5.1a. However, each of the elements of Figure 5.1b are linear with dimensional growth. "Calculate distance" to either the obstacles or the target each add only an extra term to the calculation. Also, increasing the dimension will add a term for the position and velocity within the state values; also demonstrating linear growth. Each iteration updates the state

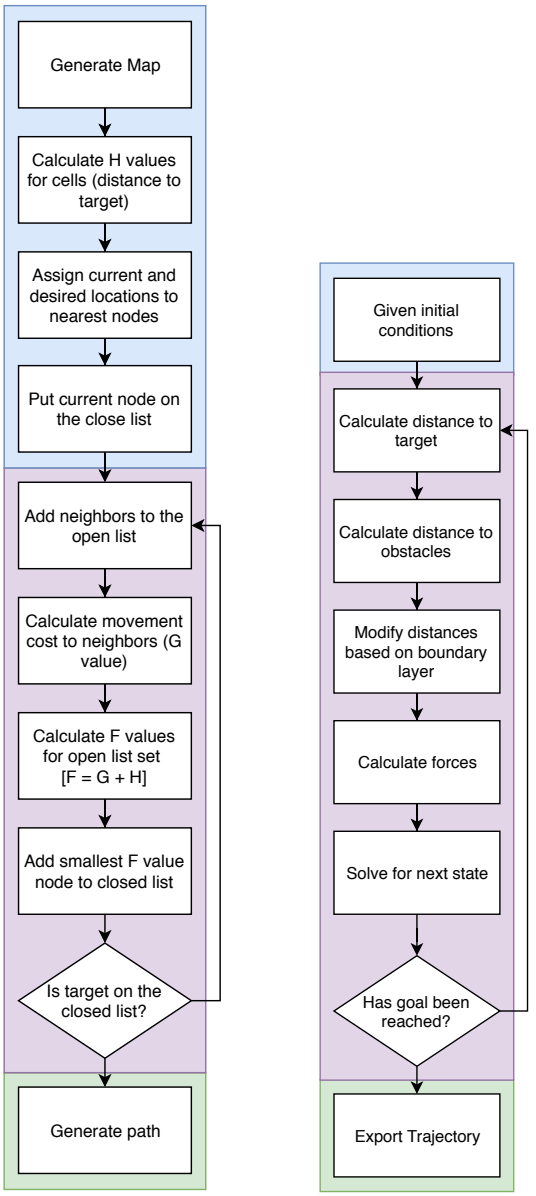
Table 5.1: Table of values for comparing A* and APF results. The locations are in n-dimensional space, and the parameters are listed to reflect that attribute. The table also lists the algorithms that the parameters are relevant to: General, both algorithms, or the either the A* or APF algorithm. Each of the values are in generalized units.

Algorithm	**Parameter**	**Value**
General	Start Location	(1, 1, ... 1)
	Goal Location	(0, 0, ... 0)
	Object Location	(0.51, 0.5, ... 0.5)
	Object Radius	0.1
A*	Grid Resolution	Variable
	Move Cost	0.7071
APF	Attraction Constant	200
	Repulsion Constant	0.01
	Damping Constant	100
	Time Step	0.001
	Boundary Layer Radius	0.2

of the system, and each update can be shown to be linearly dependent on the dimension of the system. A caveat in this process: each iteration is also linearly dependent on the number of obstacles in the system. Whereas with A* and other search algorithms, the obstacles are only relevant in determining impassable paths, the APF algorithm needs to know the relative distance between the current position and the obstacle at each iterative step. However, while each additional obstacle increases the calculation time for each iteration, this growth is also linear. Examining the flow-charts for each algorithm should demonstrate why the methods grow either linearly or exponentially. The A* method has to search in each additional dimension. The APF methods only need to account for an additional term with each new dimension.

5.3 Growth with Precision

An often overlooked component of the A* algorithm process is the map generation. Creating a map will be specific to the application and must sufficiently define the space so that the A* algorithm can navigate from the initial conditions to the goal. Thus, the number of nodes in the map also defines the accuracy of the algorithm. If the target, starting location,



(a) General A* algorithm (b) General APF algorithm

Figure 5.1: Comparison of A* and APF algorithms. In these flowcharts, the blue highlighted sections are the setup, the purple sections are iterative, and the green sections are the output. Several steps in the A* algorithm, including "Generate Map" and "Add neighbors to the open list" expand exponentially with dimension, while each element of the APF algorithm will only expand linearly with dimension.

or obstacles do not exactly lie on a node, then the map will be an approximation. The higher level of precision in the map, the more iterations that will be required to navigate. This growth is not the same as the growth with dimension, but it does indicate the non-linear nature of growth within the algorithm, and it has been included in this examination since mapping higher dimensional spaces is hard to visualize.

All of the experiments mentioned in this section were performed using the parameters from Table 5.1. The map is intentionally simple with only one obstacle to avoid. The purpose of these examinations is not to demonstrate the effectiveness of the theories but to show how the number of iterations and time to convergence changes under varying circumstances. To show how the number of iterations required for convergence grows with cell density, the field was mapped using A* using a range from 10 nodes per map length to 150 nodes per map length, and the number of iterations required to converge on the target location was measured. The results are shown in Figure 5.2 with Figure 5.3 showing several of the maps used in the experiment. The growth demonstrated in Figure 5.2 is clearly non-linear. Essentially, as the density of nodes increases, the "surface area" of open nodes surrounding closed nodes increases. To reiterate, this is not exponential growth; this experiment was performed since two dimensions is convenient to visualize, but it also demonstrates non-linear growth. In comparison, Figure 5.4 shows the Secant Method's solution to the setup. Unlike A*, a mapping process was not required, and as such, each point on the trajectory devised by the algorithm is to computational precision. Since there is no cell density, their is no growth comparison with cell density that can be shown, but the generated path is included in this section to demonstrate that the system is solvable at much higher precision possible than A* methods would allow.

The rate of growth with precision is entirely dependent on the nature of the algorithm. Within each iteration of the A* method, a node changes its designation form the "open" to the "closed" list. Each open node is a neighbor of a previously discovered closed node. Thus, the purpose of the algorithm could be defined as methodically transferring nodes

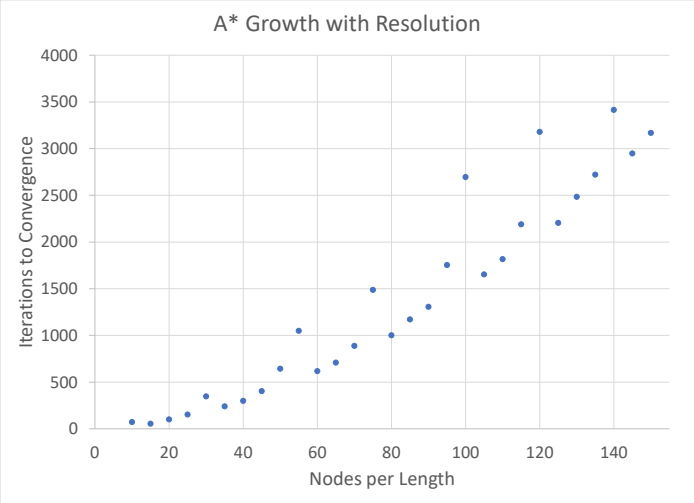
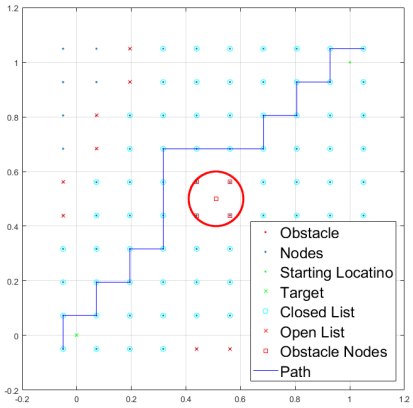
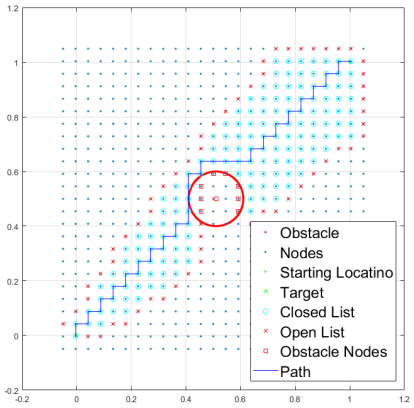


Figure 5.2: Figure showing the number of iterations to convergence based on the resolution of nodes within the map using the A* algorithm. This expansion is not the same as the exponential growth with dimension, but it is a demonstration of the non-linear nature of growth within the map.

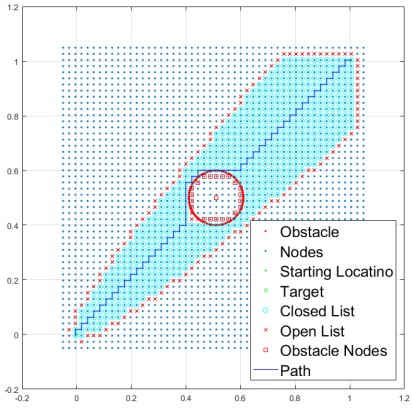
from open to closed in a meaningful way that will converge on the node closest to the target. As such, since the open nodes list essentially forms the perimeter of the closed region, non-linear growth is expected as density of nodes increases. In contrast, the APF approach does not rely on a map with a density of cells, but rather relies on the numeric precision of the processor calculating the trajectory. Thus, the Secant approach will always be performed at the highest level of accuracy possible given the processor available. Although this growth is not the same as the rate of growth with dimension, it is an important consideration when examining methods. The number of iterations required to converge using A* grows drastically the more precision that is required, while the APF methods can be as accurate as computational and numeric precision allows.



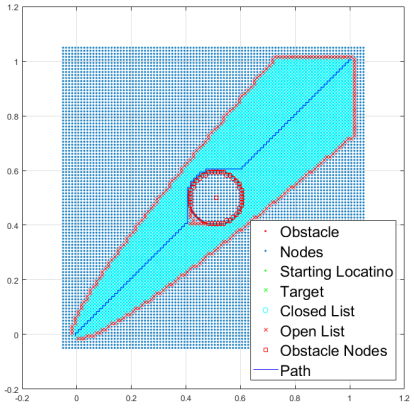
(a) 10 Nodes per length



(b) 25 Nodes per length



(c) 50 Nodes per length



(d) 100 Nodes per length

Figure 5.3: Solution to the map at various resolution sizes using A*. The denser the grid of nodes, the higher the accuracy in the path, but the greater the number of iterations to generate the path.

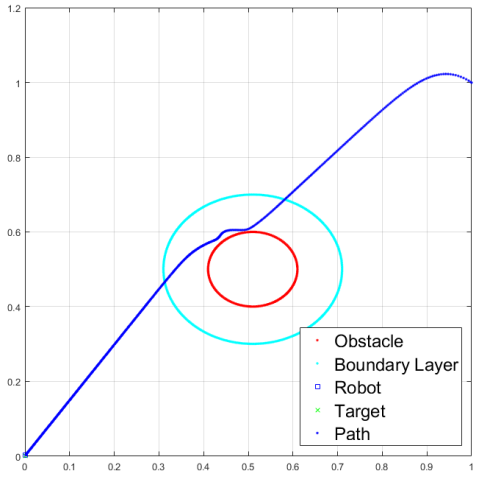


Figure 5.4: Solution to the map using the Secant approach. Note: the solution is not necessarily optimal, but it generates a trajectory with machine precision.

5.4 Growth with Dimension

The critical experiment of this chapter is to map the growth rate of the A* algorithm with dimension. This experiment is complicated by one very important consideration: the size of the map necessary for A* grows considerably fast, even with extremely low density of cells. With this restriction in mind, the path was mapped with the A* algorithm at up to five dimensions for resolutions of 4, 5, and 6 nodes per length of the map. The number of iterations and the time to convergence is depicted in Table 5.2 and shown graphically in Figure 5.5. In the previous section, it was noted that the growth of the A* method with the density of cells is not exponential; however, the growth of A* with dimension is exponential. Clearly, the results shown in Figure 5.5 are quite unimpressive. With only six cells per length of the map, at five dimensions, the system took over 4 minutes to solve. Thus, for systems greater than a few dimensions, A* will not be able to perform with necessary accuracy without drastically large maps.

The Secant approach was also performed on the multi-dimensional map. However, unlike the A* algorithm, it is not necessary to map the space before the algorithm can be

Table 5.2: Table demonstrating the growth in convergence for both iterations and time using the A* algorithm. Notice, the resolution for each of these maps is extremely low, with only a few nodes per length. This restriction in resolution is due to the computation time at higher resolutions at higher dimensions as well as the computational cost of storing the high dimensional maps.

Resolution	Dimension	Iterations	Time to Converge (sec)
4 Nodes per Length	2	12	0.05
	3	58	0.07
	4	248	0.39
	5	1014	4.48
5 Nodes per Length	2	16	0.05
	3	109	0.19
	4	600	1.57
	5	3109	41.24
6 Nodes per Length	2	26	0.02
	3	201	0.19
	4	1272	6.10
	5	7741	249.50

solved. Also, the Secant approach does not rely on searches to perform its trajectory planning. Thus, the scaling of dimension with the Secant approach is much less drastic than with A*. Figure 5.6 shows the number of iterations to convergence compared with dimensions for the Secant Method. The growth rate of the number of iterations to convergence is actually slower than linear. However, the time to completion for each iteration is also dependent on dimension, so the time to convergence was also plotted, as shown in Figure 5.7. This series clearly shows a linear growth rate, as expected since each step within the algorithm is linearly dependent on the number of dimensions.

It has not been the purpose of this analysis to directly compare the time to completion of the A* method and the Secant Method. Different optimization techniques and coding strategies could improve either algorithm. However, it is important to note the limitations placed on this analysis with A*'s growth rate. After five dimensions, calculating the map of A* became impractical, let alone applying the algorithm for generating a path. In contrary, the Secant approach was able to find the trajectory in over 100 dimensional space without

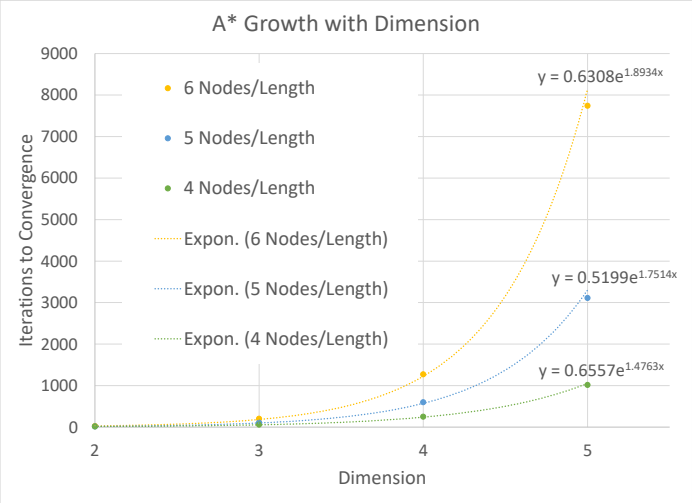


Figure 5.5: Figure depicting the results from Table 5.2. In this figure, the best-fit lines are shown as exponential equations.

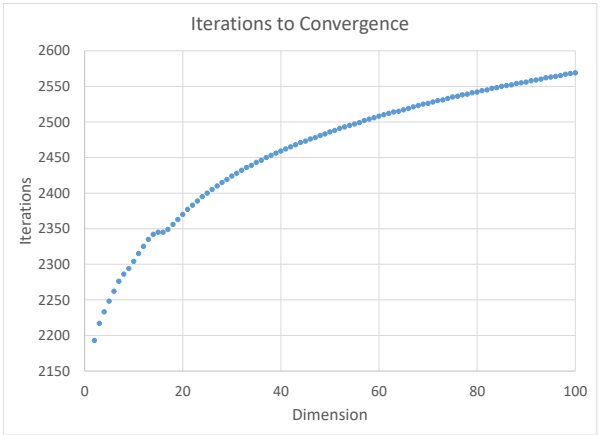


Figure 5.6: Figure depicting the number of iterations to convergence for the Secant Method. This graph is a bit misleading, since each iteration is dependent on the dimension of the system. This graph is obviously non-linear, but instead of exponentially growing, the rate of growth clearly decreases with dimension: demonstrating that as the system grows in dimension, the number of steps to convergence is not drastically increased.

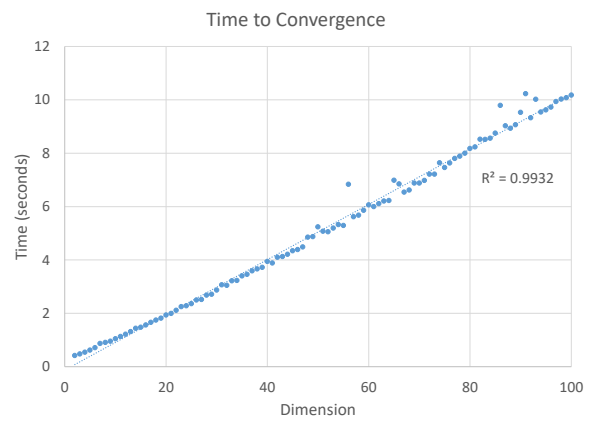


Figure 5.7: An example of the computation time for convergence using the APF Secant Method. This figure gives a better demonstration of the linearity of growth with dimension, as it encompasses the entire forward integration scheme. The computation time itself is not meant to be considered: better processors and more optimized code will reduce the computation time of any method, but rather the relative growth of the system with increased dimension.

much difficulty. A major bottle neck in path planning is working with higher-order systems, and the APF methods described are much better suited at handling these systems than search methods. With the experiments shown, the A* algorithm would not be practical for searching the space of a six degree-of-freedom manipulator with any reasonable precision, while the APF methods are more than capable of handling n-dimensional systems.

5.5 APF Growth with Obstacles

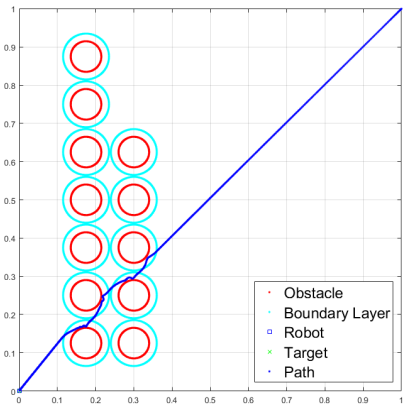
A final consideration with the APF methods that is not shared by A* and other search methods is the growth in convergence time with the number of obstacles. While fields with a greater number of obstacles may impact the convergence time of search methods such as A*, the obstacles will only have an impact on the convergence time if the obstacle is initially in the convergence path. Contrarily, Secant and TAPF approaches consider each obstacle at each iteration stage to determine the next state in the trajectory. As such, obstacles that would not seem to be in the path of convergence still affect the convergence time.

A similar experiment to the parameters listed in Table 5.1 was performed with the Secant approach. However, in this experiment, multiple obstacles were arranged in a grid-like pattern for a two dimensional system. The experiment compared the convergence time for various numbers of obstacles, from one to forty-nine; an example of these trials are shown in Figure 5.8 and the times are recorded in Figure 5.9. As Figure 5.8 shows, even though the number of obstacles grows, the convergence path does not change very much between trials.

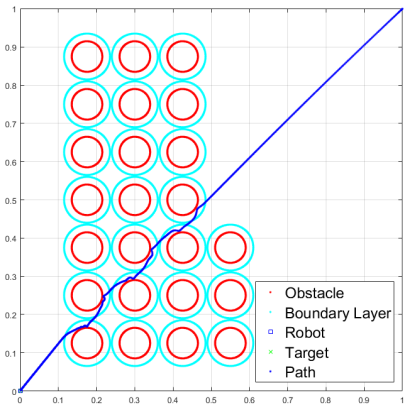
The significance of the results from Figure 5.9 is that the convergence time linearly increases with the number of obstacles. Even though the vast majority of the obstacles within the grid do not directly impact the convergence path, they are still considered within the calculations. As such, each obstacle's relative distance and artificial potential field needs to be calculated regardless of proximity. One might expect Figure 5.9 to look step-like; with the convergence time increasing only when the obstacle obstructs the convergence path, but that is not the case. However, the growth is still linear with the number of obstacles, so large numbers of obstacles can still be considered in path planning applications without fear of out-of-control growth rates.

5.6 Conclusion

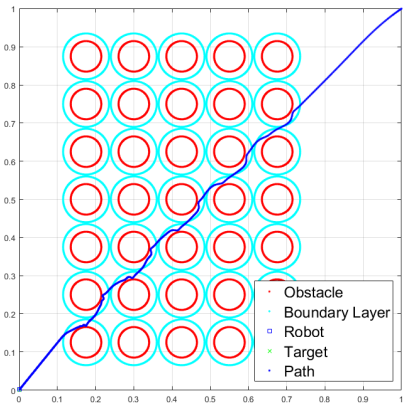
An important consideration in determining appropriate path planning algorithms is understanding how the approaches scale. In this chapter, three scaling factors have been considered: precision, dimension, and obstacle density. Precision is important in search methods that rely on mapping to define the field: low density cells may result in poor path choices, while high dimensional cells may cause unnecessary calculation. Dimension is usually defined by the problem being solved and usually cannot be altered, but it must be considered when choosing methods. Finally, knowing the obstacle density is an important consideration in evaluating the efficiency of the path planning algorithm. A* scales faster than linearly with precision and exponentially with dimension. APF methods, particularly the



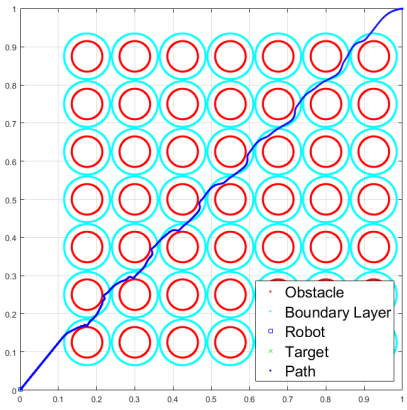
(a) Secant approach with 12 obstacles.



(b) Secant approach with 24 obstacles.



(c) Secant approach with 36 obstacles.



(d) Secant approach with 49 obstacles.

Figure 5.8: Various numbers of obstacles used to determine the time to convergent per obstacle using the Secant approach. Figure 5.9 demonstrates the linear growth in convergence time with numerous obstacles. The purpose of these figures is not to demonstrate convergence, but rather to show that the convergence time increases with each obstacle in the field, regardless of whether or not the significantly impacts the path.

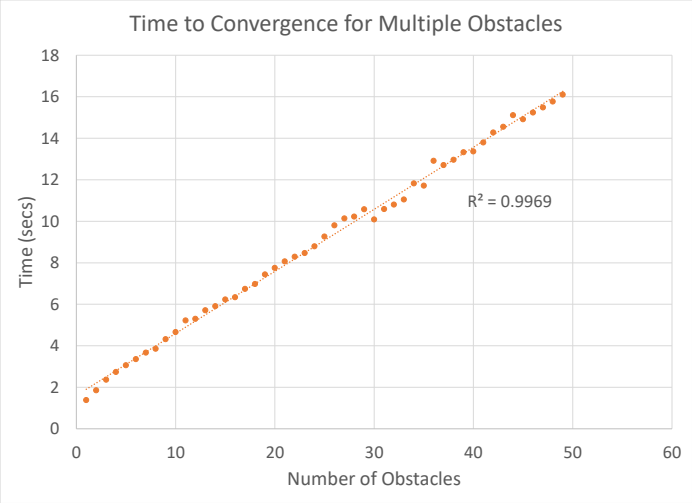


Figure 5.9: Convergence time using the Secant approach per obstacle. This graph demonstrates linear growth of convergence time versus the number of obstacles in the field.

Secant approach, scales linearly with both dimension and the number of obstacles. APF's scaling properties allow for incredibly high dimensional analysis with machine precision, a trait that A* struggles with immensely. Although the APF methods also scale with the number of obstacles, a factor that does not directly impact A*'s convergence time, that growth is also linear. While it has not been the goal of this analysis to suggest optimization techniques, an important consideration for the APF methods is that calculations could be completed with multi-threaded processors. Unlike A* and other search methods, where doubling the dimension of they system more than doubles the necessary search time, if the APF methods needed to have their dimensions doubled, doubling the processing power would lead to steady convergence rates. The performance of the APF methods is only half of their advantage, a significant draw to using these approaches is their ability to compute convergence paths for high dimensional systems.

CHAPTER 6
PLANAR MOTION WITH ROTATIONS

6.1 Introduction

The Secant and TAPF methods adequately present a way of trajectory planning in n-dimensional space. As long as the assumptions of initial conditions and obstacle configurations are met, both of these methods will converge on their target locations. However, in practice, n-dimensional path planning is often not sufficient. Many applications will involve further considerations: such as a two dimensional system with rotations. The n-dimensional planning discussed so far have been quite rigid in their approach, but adapting the system for rotation is subject to choice. Several different ways can be contrived which allows for the APF methods to be tailored for rotation, and each adaptation will have its own set of additional constraints and considerations. Overall, while the theory behind the Secant and TAPF methods are fixed: how they are utilized in specific systems must be decided upon based on the application.

As mentioned, the Secant and TAPF methods are not immediately applicable to rotations; however, a few modifications can be devised which would allow for the theories to be applied to systems with rotation. The most obvious modification for adapting a planar robot to an n-dimensional system would be to have a perpendicular axis which would represent the rotation of the robot. For this system, the radius of the robot would be grafted onto the obstacles in the field: reducing the robot to a point and increasing the size of the obstacles appropriately. The obstacles would then vary in radius based on the orientation of the robot in relation to the obstacles. The two dimensional system is now in three dimensions: with the third dimension defined by the parameters of the robot. This method of applying rotation to a system that would otherwise be dependent on point robots is not

unique, and similar methods are commonly utilized. However, for the Secant and TAPF approaches, it offers a few difficulties, which will be discussed in greater detail within the chapter, but which make the application difficult to adhere to the assumptions of these methods. Suffusive to say, other applications of the APF methods should be considered as well.

An alternative approach to accounting for rotation is to couple two, two degree-of-freedom systems by a constant length constraint. In this way, each "end" of the robotic system would be attempting to converge on its own target location while maintaining a fixed distance to the partner. Since the trajectory planning for the Secant and TAPF methods relies on solving a dynamic system, the distance constraint can be modeled as a link force holding the two ends together. Then, a single equivalent system with planar motion and rotation can be described that maintains the properties of the coupled system. This method will also have challenges and drawbacks, but it more closely utilizes the principals developed in describing n-dimensional trajectory planning with the Secant and TAPF methods.

An important consideration within this chapter is that the methods described are not the only ways of describing a planar system with rotation. This chapter is meant to examine some approaches to applying rotation to the Secant and TAPF methods and to examine the affects, but it is not meant to be exclusive. Other methods may be devised which have more benefits or greater applicability, and different scenarios may favor some methods more than others. However, this chapter will show that the Secant and TAPF approaches are suitable in handling rotational considerations as well as n-dimensional trajectory planning.

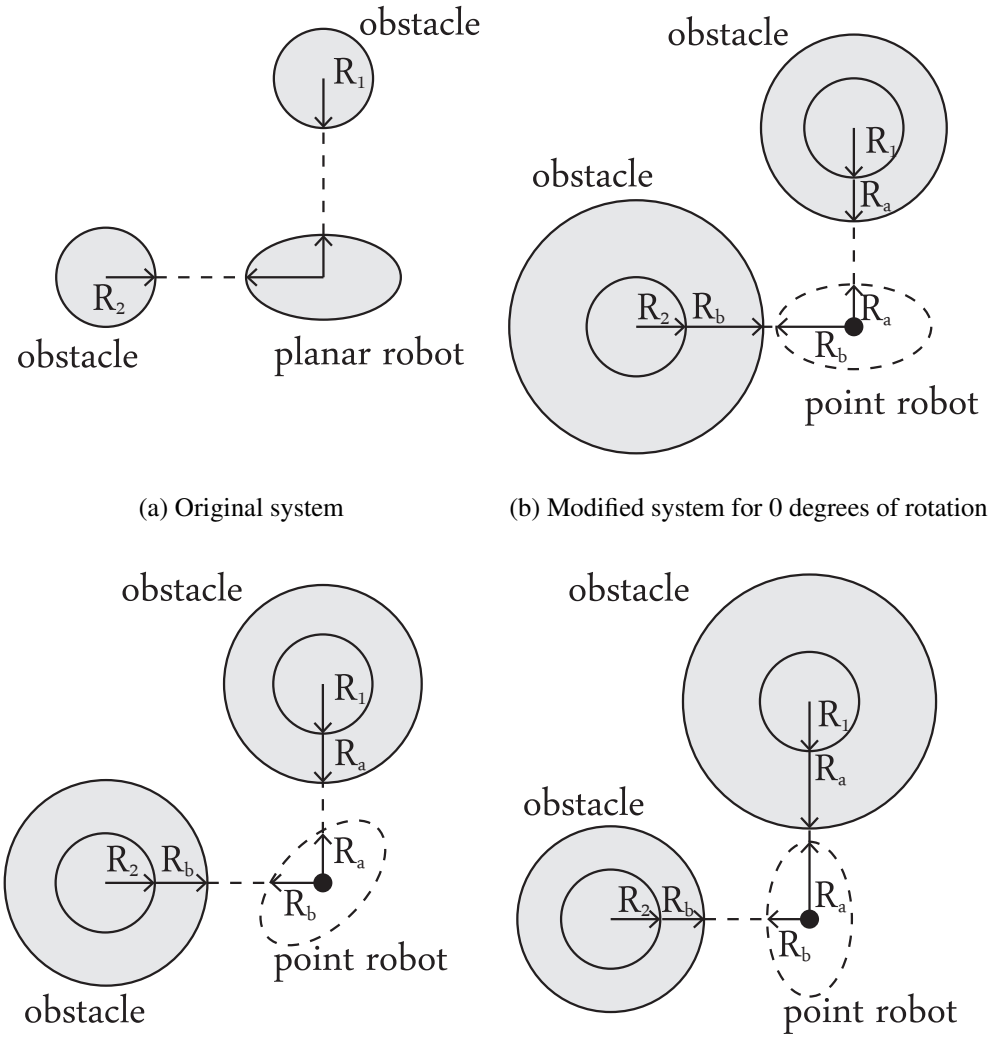
6.2 Rotation on Z-axis

The simplest way of incorporating rotation into a planar system is to create an additional degree of freedom along the z-axis which represents rotation. The robot, which is assumed to have a solid body of variable radius, is then converted to a point by adding the radius

of the robot to the obstacles: transforming the obstacles to three-dimensional structures. The radius of the obstacle now varies with height, corresponding to the grafted radius of the robot which varies with orientation. An example of how this transformation is shown in Figure 6.1. As shown, as the orientation of the robot changes, the apparent radius of the obstacles changes as well. After this transformation, the system is equivalent to a three dimensional problem and can be solved using the Secant or TAPF methods previously discussed.

Using equivalent systems, a planar robot with rotations can be transformed into an n-dimensional path planning problem. However, the Secant and TAPF approaches have some very important assumptions that may not be adhered to under these circumstances. The most obvious is the necessity to have the obstacles describable in polar coordinates. Using the variable radius of the robot to create a variable radius of the obstacle does not guarantee that the new, three-dimensional obstacle will be describable using polar coordinates. Indeed, unless the robot is purely circular, the obstacle in three dimensions will not be able to be described in polar coordinates. For each rotation of the robot, a maximum and minimum radius will be found; therefore the radius of the obstacle will oscillate with height, making the system impossible to describe with polar coordinates. Second, the height of each of these obstacles is now infinite, which violates the restriction that the obstacles must be bounded (as well as describable in polar coordinates). Finally, the obstacles cannot overlap, so the boundary of a 3 dimensional obstacles in the field must be guaranteed for all possible rotations: this requirement is not difficult to check and might be satisfied, but it must be considered. These deviations from the assumptions mean that the Secant and TAPF methods may still be applied, but they do not guarantee convergence from the starting conditions to the goal.

The theory behind the Secant and TAPF approaches guarantees convergence for point robots in solid obstacle fields of n-dimensional space so long as specific conditions are met. In adapting the theory for various applications, the assumptions must either be adhered to,



(a) Original system

(b) Modified system for 0 degrees of rotation

(c) Modified system for 45 degrees of rotation

(d) Modified system for 90 degrees of rotation

Figure 6.1: Depiction of the transformation between a planar robot and a three dimensional point robot. In the first image, the robot and obstacles both have solid dimension, but in subsequent drawings, the radius of the obstacles has changes based on the orientation and position of the robot.

or the consequences of violating the assumptions must be understood. The primary result of violating an assumption by modifying the system is that the convergence properties will no longer be guaranteed. This does not preclude convergence, but rather poses greater restrictions to the initial conditions and the layout of the field that will allow for convergence. The process of expanding the theory for systems beyond general n-dimensional path planning will always have additional considerations necessary for understanding the resulting trajectory's behavior.

6.3 2D Rotation as Constrained Systems

In order to account for a planar system with rotation, an alternative scheme could be considered with two, two-dimensional point robots with a constant length constraint. In this set-up, depicted in Figure 6.2, each point represents a two degree-of-freedom robot, and an additional constraint that defines the length between the two points creates a three degree-of-freedom system. The benefit of this set-up is that the nature of the n-dimensional trajectory planning is preserved. Each "point" (end of the robot) is treated as an individual robot, but the dynamics includes an additional force constraint in addition to the attractive and repulsive forces. Furthermore, as shown in Figure 6.3, this coupled system is dynamically equivalent to a single planar system with position and orientation.

Another important consideration is that the set-up for this system has been shown with two points creating a single robot constrained by a single length. However, it is certainly possible to expand this system for multiple points constrained in arbitrary configurations, allowing for general shapes to be constructed. Again, the benefit of modeling rotation as a constrained system of trajectories is that it more closely follows the model of n-dimensional path planning developed previously, but the impact of assumptions still needs to be considered.

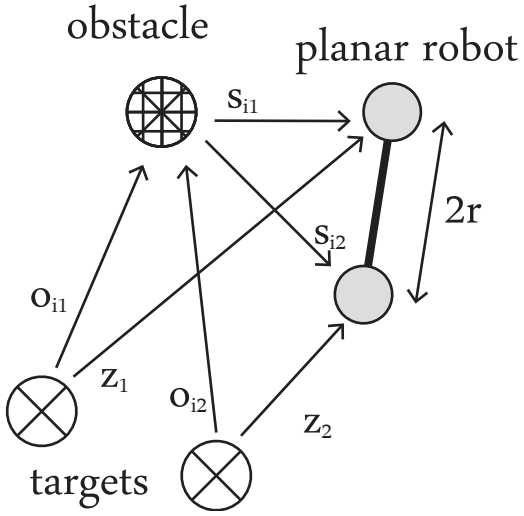
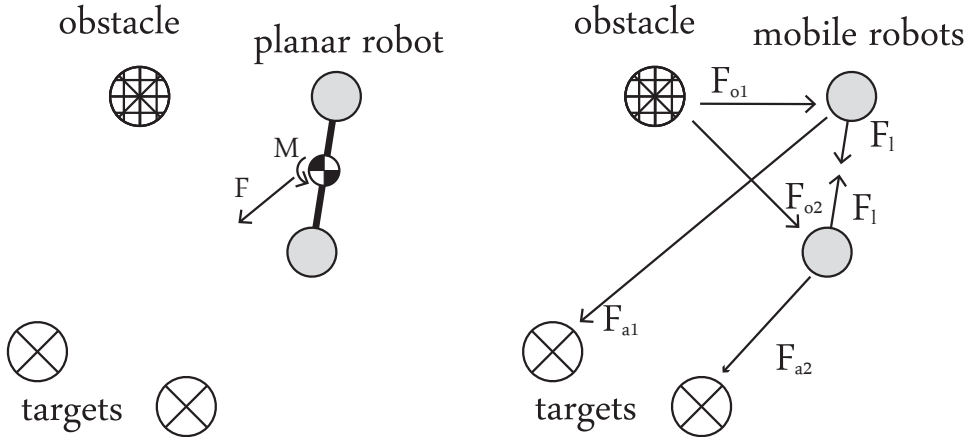


Figure 6.2: Approach for defining rotation by selecting two goal locations. Notice, each end of the robot has its own coordinate system which defines the attraction and repulsive forces.



(a) Dynamic model using position and orientation

(b) Equivalent model showing two point robots with a link force connecting them together

Figure 6.3: Two equal representations of the dynamic model. The first representation shows a single system with position and orientation, the second system is a set of points connected by a link force. The benefit of the first model is that it is easier to simulate, the second model demonstrates the stability properties.

6.3.1 Planar Motion Control

The system can be represented by the coupling of two independent point robots that are constrained by a link force which rigidly defines the distance between the points. The state equations are shown in Equations 6.1. These equations are mirrored versions of the velocity and acceleration dynamics of a single point in space, with the added component of the link force, F_l.

$$\dot{z}_1 = -\lambda_1 z_1 + v_1 \tag{6.1a}$$

$$\dot{z}_2 = -\lambda_2 z_2 + v_2 \tag{6.1b}$$

$$\dot{v}_1 = -k_{v1} v_1 - \left(\frac{\delta \Psi_1}{\delta z_1}\right)^T + F_l \tag{6.1c}$$

$$\dot{v}_2 = -k_{v2} v_2 - \left(\frac{\delta \Psi_2}{\delta z_2}\right)^T - F_l \tag{6.1d}$$

From these equations, the stability of the system can be analyzed at the collective origins of the coupled points. The energy in the system is nothing more than the combined energy of each of the points, so the positive definite Lyapunov function can be chosen as seen in Equation 6.2.

$$V = \Psi_1(z_1) + \frac{1}{2} v_1^T v_1 + \Psi_2(z_2) + \frac{1}{2} v_2^T v_2 \tag{6.2}$$

The time derivative of Equation 6.2 is shown in Equation 6.3.

$$\begin{aligned}\frac{dV}{dt} =& -\lambda_1 \left(k_{p1} + \sum_i \frac{k_{i1} \left\| o_{i1} \right\|}{\left\| z_1 - o_{i1} \right\|^3} \right) \left\| z_1 \right\|^2 - k_{v1} \left\| v_1 \right\|^2 \\ & -\lambda_2 \left(k_{p2} + \sum_i \frac{k_{i2} \left\| o_{i2} \right\|}{\left\| z_1 - o_{i2} \right\|^3} \right) \left\| z_2 \right\|^2 - k_{v2} \left\| v_2 \right\|^2 \\ & + v_1^T F_l - v_2^T F_l \end{aligned} \tag{6.3}$$

The derivative of the Lyapunov function can be shown to be negative definite. To do so, the velocity of the points can be related using Equations 6.4, where v_{com} is the velocity of the center of the joint system, r is half of the distance between the points, and ω is the rotational velocity of the joint system.

$$v_1 = v_{com} + r \times w \tag{6.4a}$$

$$v_2 = v_{com} - r \times w \tag{6.4b}$$

Implementing the velocity relationships, Equation 6.5 can be shown to be true. Remember, in this equation the direction o $r \times w$ is perpendicular to the direction of the link force.

$$v_1^T F_l - v_2^T F_l = (v_{com} - v_{com})^T F_l + 2(r \times w)^T F_l = 0 \tag{6.5}$$

Substituting the velocity relationships into the derivative of the Lyapunov equation yields Equation 6.6. This equation is the summation of time based derivatives of the independent systems, each of which is proven to be negative definite, thus the entire system

must be negative definite.

$$\begin{aligned}\frac{dV}{dt} =& -\lambda_1 \left(k_{p1} + \sum_i \frac{k_{i1} \left\| o_{i1} \right\|}{\left\| z_1 - o_{i1} \right\|^3} \right) \left\| z_1 \right\|^2 - k_{v1} \left\| v_1 \right\|^2 \\ & -\lambda_2 \left(k_{p2} + \sum_i \frac{k_{i2} \left\| o_{i2} \right\|}{\left\| z_1 - o_{i2} \right\|^3} \right) \left\| z_2 \right\|^2 - k_{v2} \left\| v_2 \right\|^2 \leq 0 \end{aligned} \tag{6.6}$$

Since the derivative of the Lyapunov equation is negative definite, the system must therefore be globally asymptotically stable at the origin.

Hopefully, the intention of modeling the system as a set of coupled points should now be clear. The Lyapunov equation is a representation of the energy of the system. However, internal forces, such as link forces, do not contribute any work on the system. Therefore, additional constraints that are modeled as internal forces should not impact the stability of the system. This insight on stability is true for a single coupled system as well as for more complex systems.

6.3.2 Additional Constraints

Modifying the system to account for two points constrained by a link force is not without consequence. In order for the system to be globally convergent, another set of considerations must be made for the initial conditions. If these conditions are not met in addition to the set of restrictions for point robots, the stability of the system cannot be guaranteed.

For point obstacles, it is essential that obstacles do not overlap in order for the system to converge. In the system of two, constrained points, each point robot has its own origin and set of obstacle positions. The system as a whole has a single origin and desired rotation, which means that the obstacle positions within the calculations might overlap. In this case, it is not guaranteed that the system will converge. This case corresponds to the robot getting "wedged" between two obstacles. If this constraint is not met, the robot may still converge on the target, but it is no longer guaranteed to converge. An additional constraint

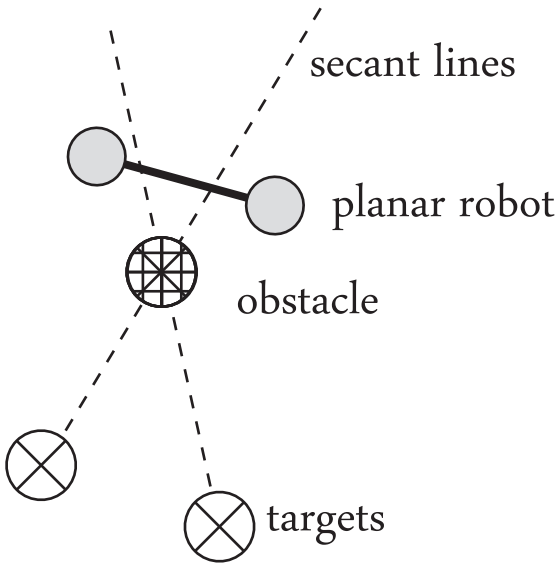


Figure 6.4: Example of an unacceptable starting condition for a coupled system. In this circumstance, the separate robots will attempt to pass on opposite sides of the obstacle to converge on the target which is not physically possible. This circumstance is guaranteed to fail for the Secant approach but may fail for the TAPF approach as well.

that is specific to the Secant method involves the initial placement of the planar robot. The robot cannot begin with its ends on opposite sides of one or multiple secant line, as seen in Figure 6.4. In this case, two circumstances are possible. If the length between points is larger than the diameter of the obstacle, the system will pass over the obstacle. If the length between points is smaller than the diameter of the obstacle, the system will not be able to pass through the obstacle. The TAPF method may also fail in this circumstance, by either getting trapped behind an obstacle or having the connecting link phase through the obstacle, but the Secant approach is guaranteed to fail in this circumstance. This limitation is partially due to the two dimensional reaction to the comet tail of the Secant approach: the space is divided behind an obstacle. Each of these cases should clearly be avoided, so further restrictions must be placed on the starting position of the coupled system and the layout of the obstacles.

These additional constraints to the system and starting conditions must be considered when applying the Secant or TAPF methods to rotation in this manner. Failure to con-

sider these constraints may result in a undesirable trajectory. However, so long as these conditions are met, the system will converge as expected.

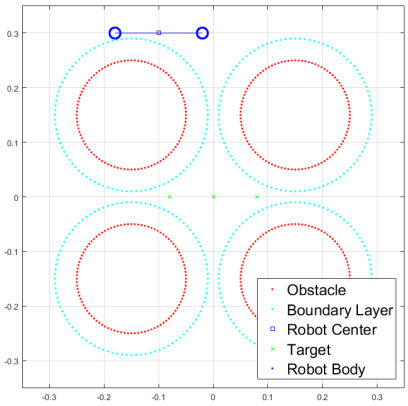
6.3.3 Planar Motion Results

Using coupled systems, so long as the restraints on the initial conditions are met, the system is guaranteed to converge. Therefore, any level of demonstration is merely a tiny subset of a much larger field of solvable systems. This section will briefly demonstrate and discuss some of the applications for which the coupled APF method could be utilized. In each of these demonstrations, the Secant approach was applied, and the system converged on the target locations. In order to demonstrate convergence, the end-points of the robot were given a solid area, and the radius of the solid was applied to the bounds of the obstacle to ensure that the system avoided intersection with the obstacles.

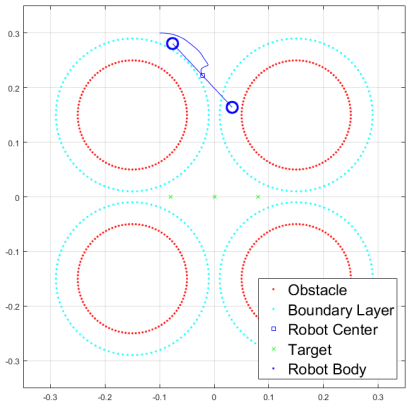
The most basic example is depicted in Figure 6.5. In this example, a planar robot (a line with orientation and position) is tasked with avoiding several solid bodied obstacles while converging on the target location. Figure 6.5 demonstrates this process of convergence over several frames from the simulation.

An extension of the basic theory is shown in Figure 6.6. Just as the point obstacle case was able to be expanded into arbitrary shapes for point robots, the same is true in the constrained system. Now, Figure 6.6 demonstrates the system avoiding squares, and Figure 6.7 goes more abstract with the system avoiding star shapes. An important observation with the figures analyzed in this section: the obstacle field layout breeches the condition necessary to guarantee convergence. Each of the obstacles are too close together, and it is possible that initial conditions could be contrived that would not allow for convergence. However, even though the systems in these scenarios do not have convergence to the target guaranteed, each system converged without incident. The success of these systems should convey the applicability of the overall theory.

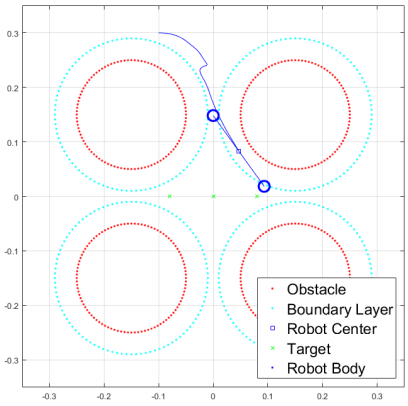
The previous examples in this section show the natural applications of using coupled



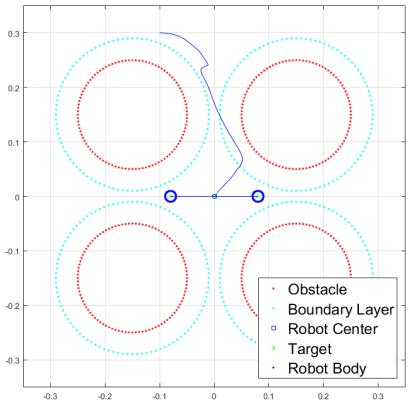
(a) Initial position



(b) One-third through path

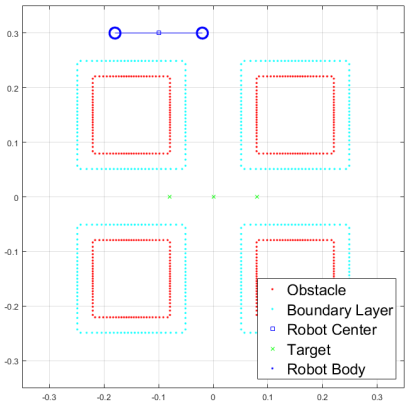


(c) Two-thirds through path

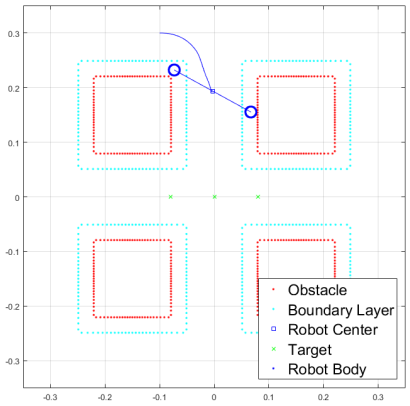


(d) Converged position

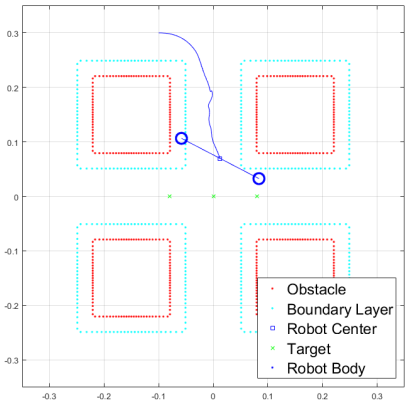
Figure 6.5: A coupled system converging on a target in a field of solid, circular obstacles using the Secant approach.



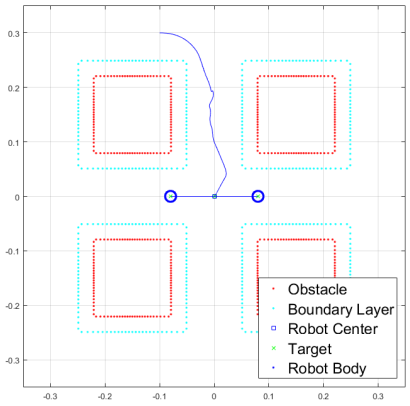
(a) Initial position



(b) One-third through path

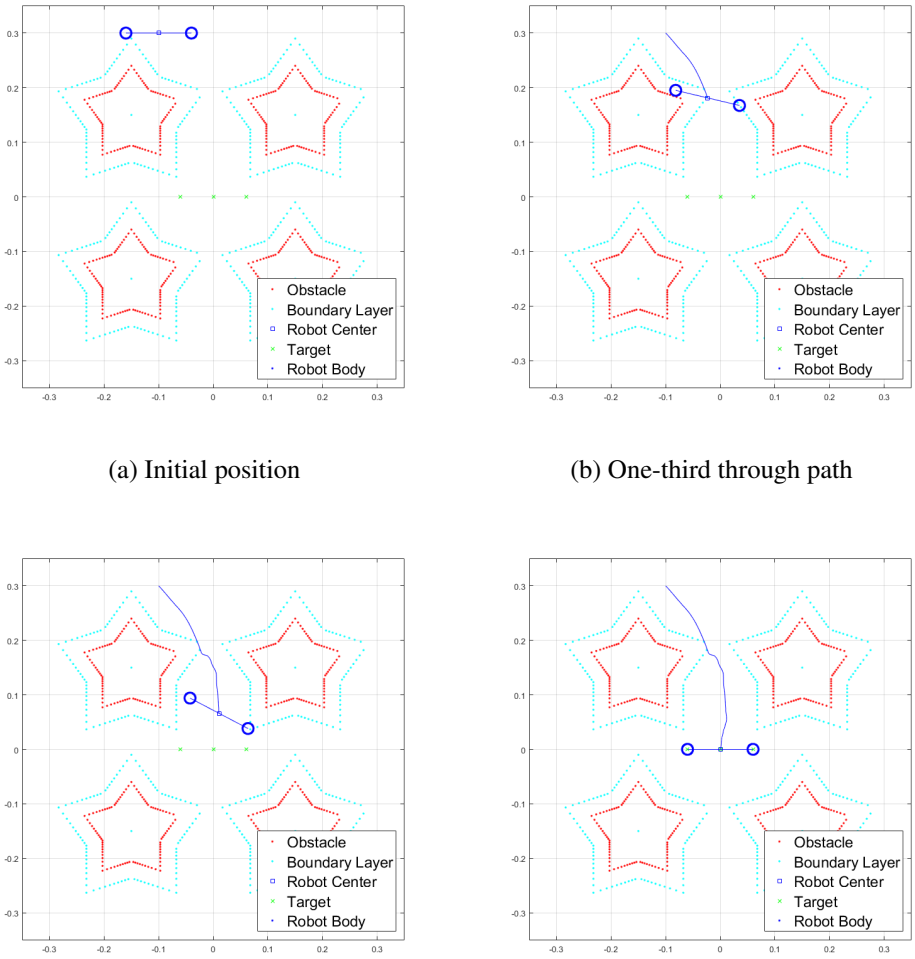


(c) Two-thirds through path



(d) Converged position

Figure 6.6: A coupled system converging on a target in a field of solid, square obstacles using the Secant approach.



(a) Initial position

(b) One-third through path

(c) Two-thirds through path

(d) Converged position

Figure 6.7: A coupled system converging on a target in a field of solid, star obstacles using the Secant approach.

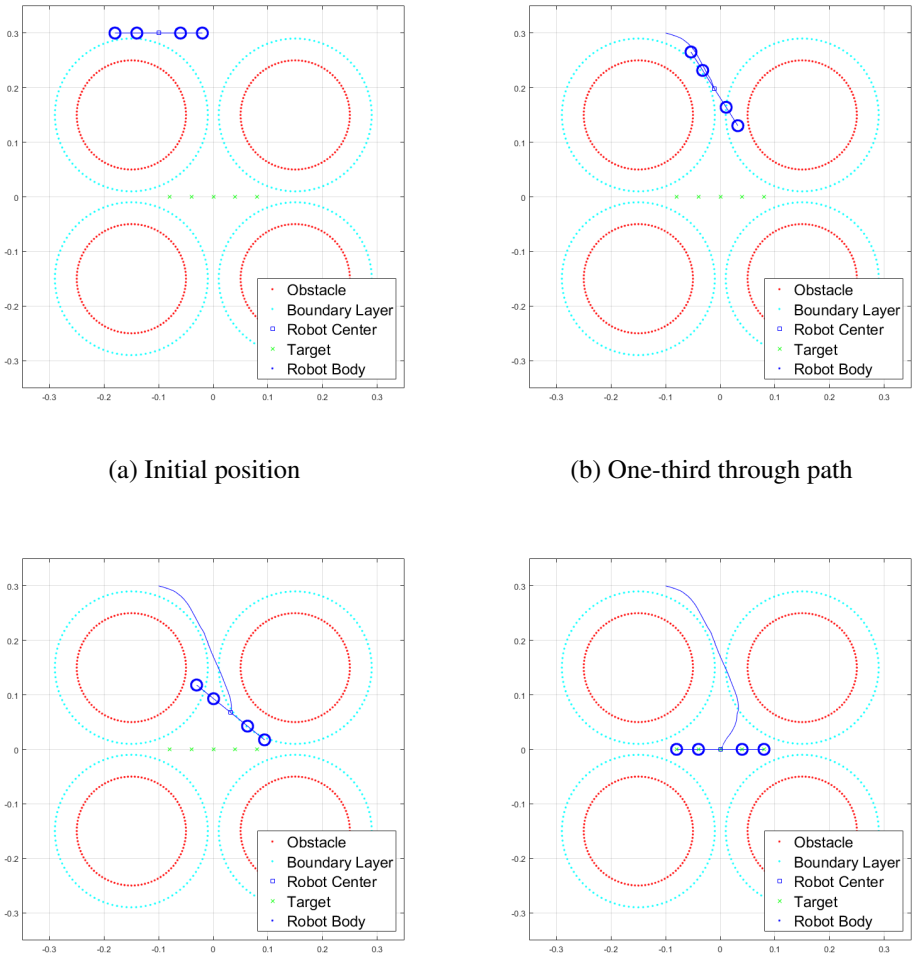
points to planar robots. However, an important consideration must be noted. The robot is only guaranteed to avoid the obstacles at the location of the points where the repellent and attractive forces are applied. Therefore, the link connecting the ends is not guaranteed to avoid the obstacles, and it may collide in practice. This problem of collision can be avoided, by introducing a series of points that each have a target location and are coupled by link forces. An example of this scenario is demonstrated in Figure 6.8. In this figure, the planar robot now has multiple points across the body of the link, making the system more robust to avoiding obstacles without loss of generality.

The final natural extension of the theory is to apply the system of coupled points to make arbitrary shapes. Just as the boundary layer approach allowed for points to appear as solid bodies, forming points into shapes interconnected by links allows for the theories developed for a point robot to be applied to arbitrarily shaped robots. An example of this application is shown in Figure 6.9, where a rectangular robot is navigating a field of square obstacles.

Each of these examples converged on their target location while avoiding the obstacles within the field. This demonstration is not meant as a comprehensive list of scenarios that the coupled-system method of trajectory planning for solid robots can handle. Indeed, these results are not even the only way in which rotation might be approached by utilizing APF methods. However, these results are meant to demonstrate that accounting for planar motion with rotation is certainly possible with APF methods, and that the properties of convergence demonstrated in previous chapters is not lost in this application.

6.4 Conclusion

The theories behind the Secant and TAPF methods were intended for a point robot with point obstacles in n-dimensional space. Using the boundary layer approach, this theory was expanded for obstacles with varying radii. Now, these theories have to be put into practice in order to describe real-world scenarios. The first application, which was examined in this



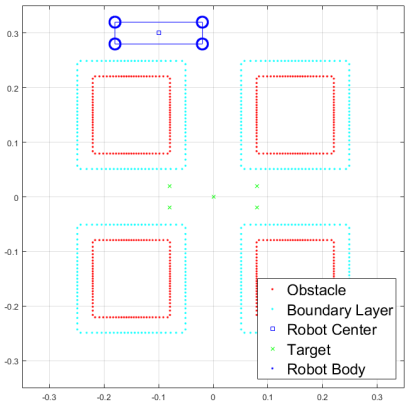
(a) Initial position

(b) One-third through path

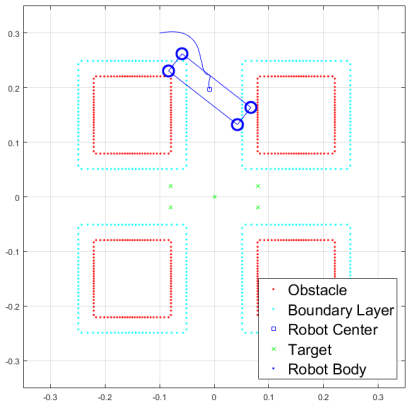
(c) Two-thirds through path

(d) Converged positions

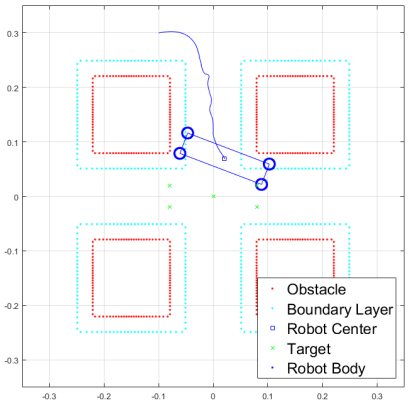
Figure 6.8: Four coupled robots converging on a target in a field of solid, circular obstacles using the Secant approach.



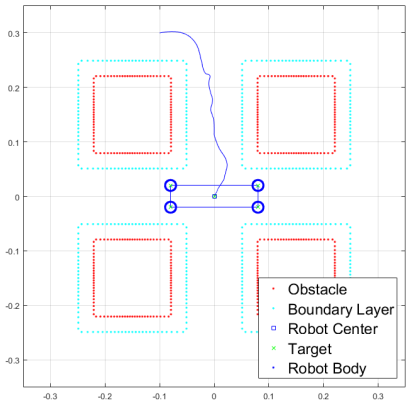
(a) Initial position



(b) One-third through path



(c) Two-thirds through path



(d) Converged positions

Figure 6.9: Four coupled robots arranged in a rectangle converging on a target in a field of solid, circular obstacles using the Secant approach.

chapter, is the concept of path planning for robots that have both position and orientation. Innumerable ways are possible for expanding the theory to account for orientation, but the method examined here was to analyze the robot as if it were a coupled system with a constraint on the length between points. If both points were to converge on the target while satisfying the constraint, then the system would be equivalent to a robot with orientation and position converging on a desired location while avoiding obstacles. This theory has several additional constraints, most concerning the layout of the field of obstacles as viewed from two perspectives. If the rectified field (the field as if the two targets were merged) does not satisfy the conditions of the point robot in a field of obstacles, then the system may not converge. However, assuming that the field does not violate these constraints, then the convergence properties of the theory hold. Furthermore, a coupled system can be expanded for a multi-point system; where link forces and points can be joined to describe arbitrarily shaped and sized robots. To be clear, this method is wholly different from the boundary layer theory, which mathematically transformed a field such that solid obstacles could be treated as if they were points. This approach of joining points is equivalent to linking multiple convergence problems simultaneously and solving them with known constraints. All of this shows a significant expansion of both the Secant and TAPF methods specifically as well as APF methods as a whole.

CHAPTER 7
MANIPULATORS

7.1 Introduction

In the previous chapter, the n-dimensional robot in a field of solid obstacles theory was expanded to account for rotational motion, in this chapter, that theory will be expanded further to account for manipulators. In many ways, robotic arms are the perfect case for where high dimensional path planning is applicable. While lower system arms certainly exist, robotic manipulators are often at least six degrees-of-freedom in order to account for position and orientation in the Cartesian space. Furthermore, higher dimensional arms are used where reach within the workspace is critical to the task. Systems like manipulators pose two very crucial challenges to path planning routines: the very nature of their high dimensional systems make path planning time consuming, and the obstacles are rarely defined in the configuration space (i.e. joint space). However, APF approaches, especially the Secant and TAPF methods with their well defined convergence properties, are well suited to not only handle high dimensional spaces, but also to account for obstacles defined within the task space. Similarly to the case with rotation, different approaches might be used to apply the theory of n-dimensional path planning, and the form of application will dictate the limitations of when the system will converge.

Planning paths for manipulators can be quite challenging. As stated previously, robotic arms are generally at least six dimensional, and the obstacles are not defined within the same space as the control routine. Therefore, path planners must be extremely processor efficient while also transforming the path into task space or the obstacles into configuration space, which in itself is quite difficult for routines to handle. Often, path planning algorithms will rely on the obstacles being perfectly defined in task space, and then will check

if corresponding points are clear within the configuration space. However, this "guess and check" approach becomes extremely costly in high dimensional systems. Fortunately, the APF methods described are well suited for operating within the task space, no matter the dimensionality of the robot arm. Whichever approach is utilized for applying the APF methods, the system can be thought of as a dynamic model with artificial forces applied to the manipulator, guiding it to the goal while avoiding the obstacles. Since this process can take place wholly in the task space, the remaining challenge is to find an appropriate way to convert the forces in the task space to joint control.

Following from the results in controlling planar motion, robotic manipulators can be viewed as a series of interconnected points with specific constraints that defines the geometry and workspace of the system. With this insight in mind, the method for controlling manipulators will begin by showing only convergence of the end-effector. Having only the end-effector avoid obstacles while converging on a target location will demonstrate an initial approach to path planning for manipulators which will then be extended for all joints within the system. This method of path planning will create new considerations of convergence that will be analyzed in detail. Similar to the analysis of planar motion, different algorithms may be contrived which will allow for APF methods to be applied to manipulators, the approaches adopted in this chapter are not meant to exclude other methods but rather it is meant as an example to demonstrate that path planning for general manipulators is possible.

Expanding n-dimensional path planning to manipulators is a challenging task. General robotic arms are controlled in their configuration space, which is generally of higher dimensionality than the task space in which they operate. Furthermore, the obstacles within the workspace need to be avoided in task space, but the path planning is often dictated in configuration space. Finally, the need for reliable transformation between the task space and the configuration space is often unavailable or extremely taxing for high dimensional robotic arms. The myriad of complications for manipulators makes path planning difficult,

but the Secant and TAPF approach are capable of handling even the highest dimensional arms with relative ease.

7.2 Control: End-Effector Only

The first extension of the n-dimensional case to manipulators will be examining only the end-effector (or terminal link) of the robot. In this scenario, only the end-effector will be avoiding obstacles while converging on a target location. The subsequent joints and links will be able to pass through obstacles and will not be convergent on any target. This simplified analysis will be used as a starting point in the analysis of manipulators and path planning. Also, this analysis will completely ignore gravity. Adding gravity to the dynamics of the manipulators can be done when appropriate, but it will not greatly change the convergence criteria. An important consideration in this analysis is that the APF methods are crucial to help the system converge, but the geometry of the arm itself also plays a role in whether or not the end-effector will be able to reach the target. Obvious exclusions to convergence, such as if the goal is outside of the manipulator's workspace, will exist and will be examined within this analysis. Another consideration: the equations developed are for generalizable "revolute" arms, where the actuators are all rotational, but the theory could easily be extended for linear actuators as well. The "end-effector only" case is meant to demonstrate that globally convergent APF methods can be used to bridge the gap between configuration and task space path planning.

7.2.1 Trajectory Planning

The dynamics of a generalized manipulator can be seen in Equation 7.1. In this equation, q is the joint state of the system, τ is the control parameter and the applied motor torque, $H(q)$ is the symmetric mass matrix used to describe the system, and $C(q, \dot{q})$ is the coriolis

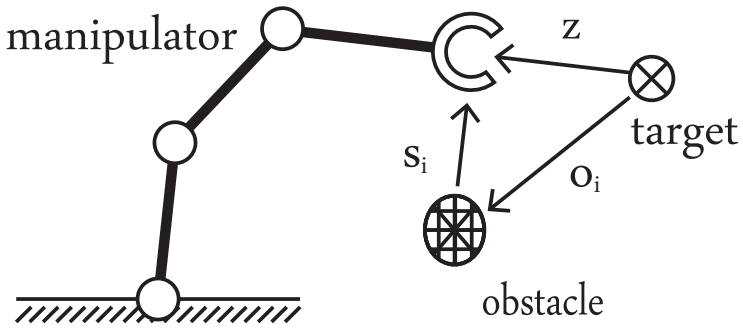


Figure 7.1: Manipulator with the end-effector converging on a target location while avoiding obstacles. For this set-up, subsequent joints may intersect obstacles and will not converge on a target location.

matrix of the system. The system is depicted in Figure 7.1.

$$H(q)\ddot{q} + C(\dot{q}, q)\dot{q} = \tau \tag{7.1}$$

The torque applied to the system can be accomplished using either the Secant or TAPF methods; however, this analysis will assume that the Secant approach was used, shown in Equation 7.2. In this equation, J is the Jacobian of the arm at the end-effector, and K_D is a diagonal matrix of dampening values, where each value is positive.

$$\tau = -J^T \begin{bmatrix} \frac{\partial \Psi(z)}{\partial z}^T \\ \vec{0}^T \end{bmatrix} - K_D \dot{q} \tag{7.2}$$

An important note in Equation 7.2, the matrix $\begin{bmatrix} \frac{\partial \Psi(z)}{\partial z}^T \\ \vec{0}^T \end{bmatrix}$ is made of two components: the gradient of the potential field and a zero vector. The zero vector is meant to account for the torsional affects in the Jacobian transpose. The way that this system is meant to converge, only the forces from the target and obstacles are considered. Equation 7.2 is also quite significant for its role in conversion between the configuration and task space. All of the obstacles considered are in the task space of the robot, and the artificial forces are

therefore computed in that same space. Furthermore, the Jacobian transpose is a forward transformation between the two spaces, which means that the Jacobian inverse is never required. Thus, the manipulator can be of arbitrary dimension and size without loss of generality in the convergence. Finally, with this torque input, the origin of the field (which is not necessarily the origin of the manipulator) is a stable point, since the gradient of the potential field is zero at the origin, and thus the acceleration of the joints is also zero.

To demonstrate the convergence criteria, the Lyapunov function given in Equation 7.3 will be examined. This equation is a representation of the kinetic and potential energy of the system and is clearly positive definite. The same bounds on the invariant set Ω_ℓ will be placed as originally outlined in Section 2.5, indicating that the initial conditions cannot coincide with the obstacle or the comet tails.

$$V = \frac{1}{2}\dot{q}^T H \dot{q} + \Psi(z) \tag{7.3}$$

An important insight is that the power input is equal to the change in kinetic energy of the system, shown in Equation 7.4.

$$\frac{1}{2}\frac{d}{dt}\left[\dot{q}^T H \dot{q}\right] = \dot{q}^T \tau \tag{7.4}$$

Taking the derivative with respect to time of the Lyapunov function and substituting values for power yields Equation 7.5.

$$\frac{dV}{dt} = \dot{q}^T \tau + \frac{\partial \Psi(z)}{\partial t}\dot{z} \tag{7.5}$$

At this point, in order to transform the system into a single domain (instead of having mixed joint values and Cartesian values), the gradient will be modified to account for a rotational component, ω; however, this value will always be multiplied by a zero vector so

it has no impact on the overall system, as seen in Equation 7.6.

$$\frac{dV}{dt} = \dot{q}^T \tau + \begin{bmatrix} \frac{\partial \Psi(z)}{\partial t} & \vec{0} \end{bmatrix} \begin{bmatrix} \dot{z} \\ \omega \end{bmatrix} \tag{7.6}$$

Completing the transformation between spaces, the velocity of the end-effector is related to the velocity of the joints by the relationship seen in Equation 7.7.

$$\begin{bmatrix} \dot{z} \\ w \end{bmatrix} = J\dot{q} \tag{7.7}$$

Substituting values into the Lyapunov derivative yields Equation 7.8.

$$\frac{dV}{dt} = \dot{q}^T \left[-J^T \begin{bmatrix} \frac{\partial \Psi(z)}{\partial z}^T \\ \vec{0}^T \end{bmatrix} - K_D \dot{q} \right] + \begin{bmatrix} \frac{\partial \Psi(z)}{\partial t} & \vec{0} \end{bmatrix} J\dot{q} \tag{7.8}$$

Equation 7.8 can be simplified into Equation 7.9.

$$\frac{dV}{dt} = \dot{q}^T \left[-J^T \begin{bmatrix} \frac{\partial \Psi(z)}{\partial z}^T \\ \vec{0}^T \end{bmatrix} + J^T \begin{bmatrix} \frac{\partial \Psi(z)}{\partial z}^T \\ \vec{0}^T \end{bmatrix} - K_D \dot{q} \right] \tag{7.9}$$

Then, Equation 7.9 can be further simplified into Equation 7.10 which is negative semi-definite.

$$\frac{dV}{dt} = -\dot{q}^T K_D \dot{q} \leq 0 \tag{7.10}$$

7.2.2 Convergence Criteria

Having Equation 7.10 be only semi-negative definite rather than negative definite is a significant divergence from previous theorems. In order to determine the convergence criteria, the Invariant Set Theorem can be utilized, in which the set of all systems for which $\dot{V} = 0$ is analyzed. In this case, if the derivative of the Lyapunov equation is zero, then $\dot{q}$ must

also be equal to zero. Therefore, Equation 7.11 must be satisfied for all stationary points within the invariant set.

$$H\ddot{q} = -J^T \begin{bmatrix} \frac{\partial \Psi(z)}{\partial z}^T \\ \vec{0}^T \end{bmatrix} \tag{7.11}$$

Of course, for a stationary point, $\ddot{q}$ must also be zero, so Equation 7.12 must be true.

$$-J^T \begin{bmatrix} \frac{\partial \Psi(z)}{\partial z}^T \\ \vec{0}^T \end{bmatrix} = \vec{0} \tag{7.12}$$

Equation 7.12 will be true under two circumstances, when $z = 0$, indicating that the target has been reached, or if the gradient of the potential is in the null space of the Jacobian transpose. The physical intuition of this condition is if the forces acting on the robot due to the obstacles and the target do not apply any torque to the joints. This may occur in circumstances where the target is reachable by the arm, but it will certainly occur if the target is outside of the workspace of the arm. These respective cases are illustrated in Figures 7.2 and 7.3. This position in which the convergence is not guaranteed is not a result of the APF methods used but rather in how the APF method is applied to the system. The arms have physical limitations, and this approach is simulating virtual forces on a physical system. It is only natural that limitations would be revealed from an analysis of the underlying mathematics.

7.2.3 Results

The results in this section will focus on high-degree of freedom manipulators comprised of revolute joints arranged in a plane. These robots are often referred to as "snake-like" due to their flexibility. The results are shown for the planar case since they are more easily documented, but the equations are generalizable to non-planar robots as well. Results in this section used the Secant approach and only the end-effector is converging to a target

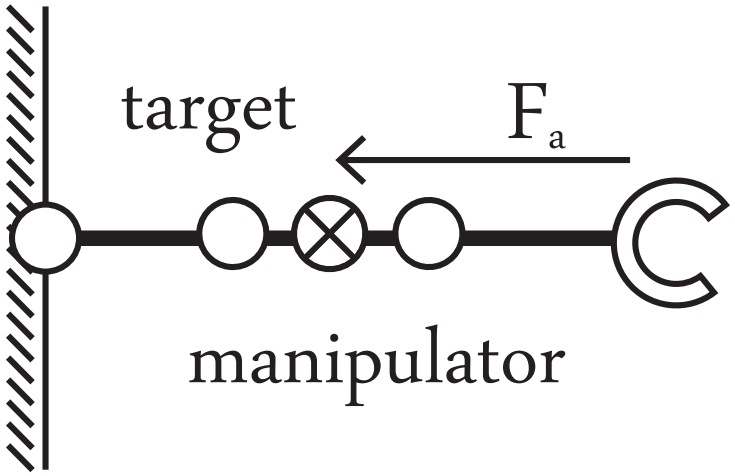


Figure 7.2: A circumstance in which the target location is within the null space of the Jacobian transpose. In this image, a manipulator with three revolute joints is attempting to converge the end-effector on a target, but the attractive force F_a applies no torque to the system.

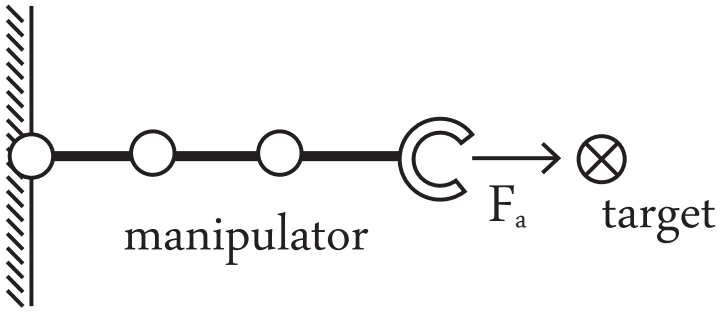


Figure 7.3: A circumstance in which the target location is outside of the workspace of the arm. This system will obviously never converge on the target location regardless of the initial conditions of the arm.

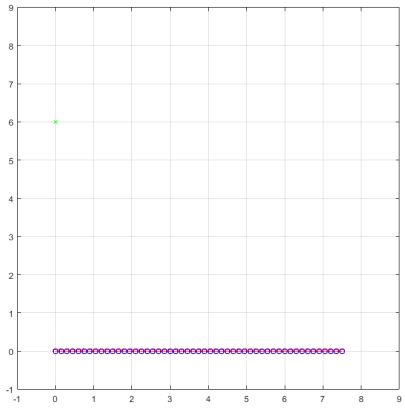
location while avoiding obstacles. Figure 7.4 first demonstrates this convergence technique in a field without obstacles. In Figure 7.4, a fifty link manipulator converges on a target location. This high dimensional application was possible due to the use of the APF approach. Extending the theory, Figure 7.5 introduces solid, circular obstacles that the end-effector of a twenty link, snake-like robot avoids while converging on a target location. Congruent with expectations, the subsequent joints pass through the obstacles as the end-effector converges. Figures 7.4 and 7.5 are merely demonstrations of the Secant approach applied to manipulators; however, the theory must be extended for consideration of all the joints within the system.

7.3 Control: All Joints

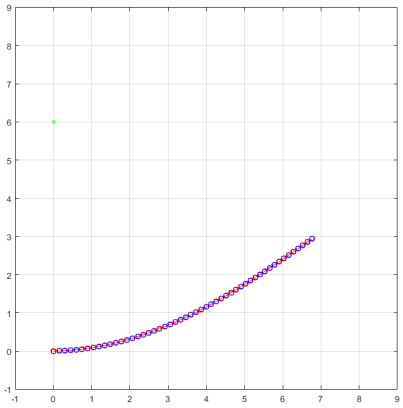
Expanding the results for the end-effector to include obstacle avoidance for all joints is a significant advancement of the theory. Several major considerations must be held: not every point which is meant to avoid obstacles will have a goal position; conditions similar to the end-effector only scenario may arise in which the forces will align with the Jacobian transpose; and also, APF methods are subject to structural singularities which will appear in any mathematical model. Understanding these potential pitfalls, the results of the planar rotational model will avail the manipulator case: the end-effector will still converge to a target location, but subsequent joints will also experience repellent forces due to obstacles. In this way, the system is designed to converge on a target while ensuring that the links will not collide with obstacles in the field.

7.3.1 Trajectory Planning

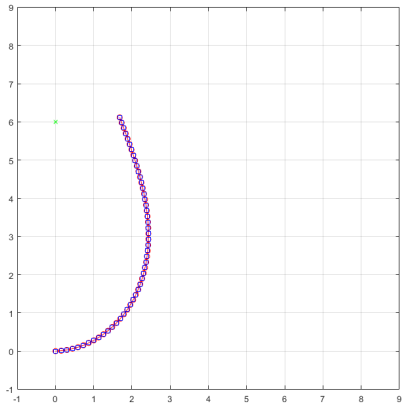
In order to analyze the system in which all of the joints converge towards a target while avoiding obstacles, the Jacobian of the robotic system will be broken up into column vectors, as seen in Equation 7.13. In these equations, the subscript n denotes the number of



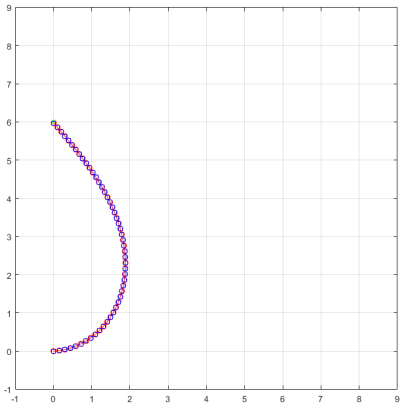
(a) Fifty link, snake-like robot, initial condition



(b) One-third through its path

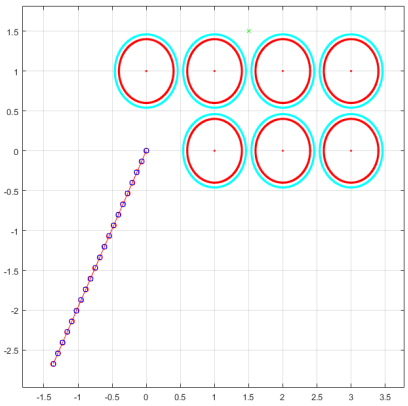


(c) Two-thirds through its path

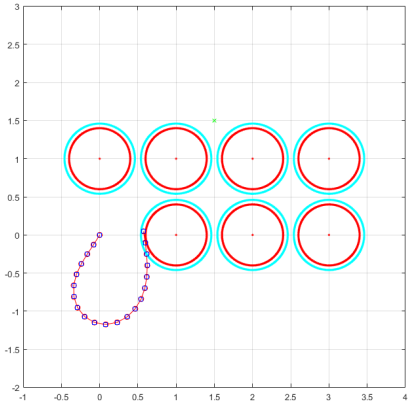


(d) Converged on target

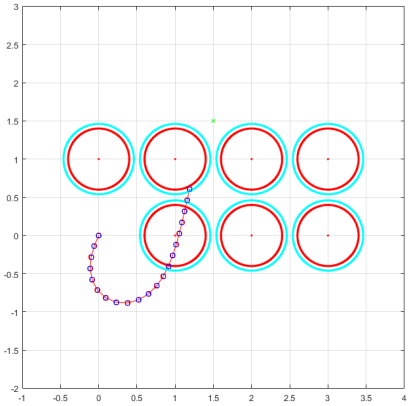
Figure 7.4: Demonstration of a fifty link robot arm comprised of revolute joints converging on a target location with a fixed base located at [0, 0]. In this example, the end-effector is the only part of the robot that experiences the potential field and there are no obstacles.



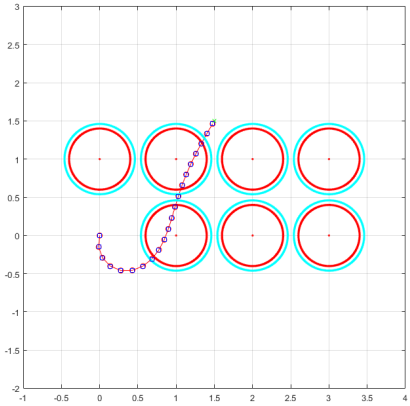
(a) Twenty link, snake-like robot, initial condition



(b) One-third through its path



(c) Two-third through its path



(d) Converged on target

Figure 7.5: Demonstration of a twenty link robot arm comprised of revolute joints converging on a target location through a field of solid, circular obstacles with a fixed base located at [0, 0]. In this example, the end-effector is the only part of the robot that experiences the potential field.

joints in the robot.

$$J = [E_1, E_2, ...E_n] \tag{7.13}$$

Then, a partial Jacobian, J_i, which is populated with columns up to the i'th joint from the Jacobian, and the rest are zero vectors, as seen in Equation 7.14. This will assist in transforming each of the forces on the joints from task space to configuration space.

$$J_i = [E_1, E_2, ...E_i, \vec{0}, \vec{0}, ..., \vec{0}] \tag{7.14}$$

The system dynamics are the same as they were in Equation 7.1, except now the applied torque has to account for the forces on each of the joints. At this point, a decision must be made as to how the system will be controlled. As with all examples so far, the application is subject to choice and will affect how the system behaves and its convergence properties. For this analysis, the end-effector will be controlled using the Secant approach, following from the previous section, and a force will be applied to subsequent links based on the TAPF approach, developed in Equation 3.6. In these equations, $\Psi(z)$ is the Secant Approach potential experienced by the end-effector. The control input is therefore seen in Equation 7.15.

$$\tau = -K_D\dot{q} - J^T \begin{bmatrix} \frac{\partial \Psi(z)}{\partial z}^T \\ \vec{0}^T \end{bmatrix} - \sum_{i}^{n-1} J_i^T \begin{bmatrix} (F_T)_i \\ \vec{0}^T \end{bmatrix} \tag{7.15}$$

The Lyapunov equation is therefore similar to Equation 7.3, as shown in Equation 7.16

$$V = \frac{1}{2}\dot{q}^T H \dot{q} + \Psi(z) \tag{7.16}$$

Using relationships between the Cartesian and configuration velocity of the joints, the

derivative of the Lyapunov equation with respect of time is shown in Equation 7.17.

$$\frac{dV}{dt} = \dot{q}^T \left[-K_D \dot{q} - J^T \begin{bmatrix} \frac{\partial \Psi(z)}{\partial z}^T \\ \vec{0}^T \end{bmatrix} - \sum_i^{n-1} J_i^T \begin{bmatrix} (F_T)_i \\ \vec{0}^T \end{bmatrix} \right] + \begin{bmatrix} \frac{\partial \Psi_i(z)}{\partial z} & \vec{0} \end{bmatrix} J\dot{q} \tag{7.17}$$

Similar to Equation 7.8, 7.17 can be shown to be semi-negative definite, as seen in Equation 7.18 and Equation 7.19.

$$\frac{dV}{dt} = -\dot{q}^T \left[K_D \dot{q} - \sum_i^{n-1} J_i^T \begin{bmatrix} (F_T)_i \\ \vec{0}^T \end{bmatrix} \right] \tag{7.18}$$

$$\frac{dV}{dt} = -\dot{q}^T K_D \dot{q} - \sum_i^{n-1} v_i^T (F_T)_i \leq 0 \tag{7.19}$$

In Equation 7.19, the terms $v_i^T (F_T)_i$ can be shown to be less than or equal to zero by the same analysis used in Chapter 3 beginning at Equation 3.12.

7.3.2 Convergence Criteria

Similar to Equation 7.10, Equation 7.19 is only semi-negative definite, indicating that there are situations in which the system would not converge on the target location. This circumstance is not unexpected, the end-effector only case would not converge if the potential gradient was in the direction of the null space of the Jacobian transpose. However, this case is more complex. As seen in Equation 7.20, further conditions may result in the system reaching a stable point that is not the target location.

$$H\ddot{q} = -J^T \begin{bmatrix} \frac{\partial \Psi(z)}{\partial z}^T \\ \vec{0}^T \end{bmatrix} - \sum_i^{n-1} J_i^T \begin{bmatrix} (F_T)_i \\ \vec{0}^T \end{bmatrix} = \vec{0} \tag{7.20}$$

One of the difficulties in locating conditions in which the manipulator will not converge is in attempting to account for structural singularities within the system. Structural

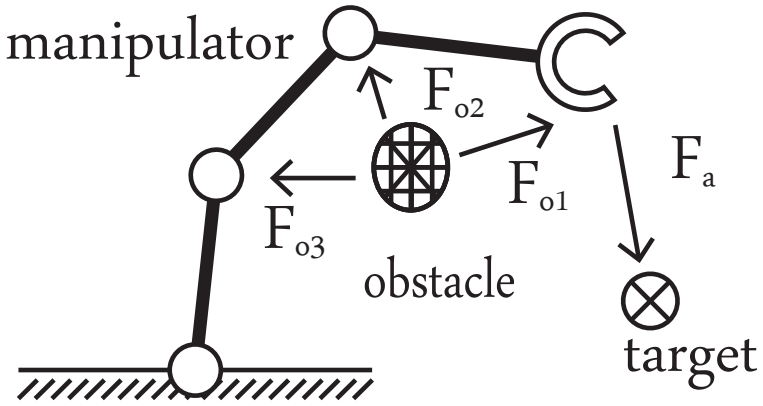


Figure 7.6: Demonstration of a structural singularity. In this scenario, the obstacle blocks the end-effector from reaching the target location; however, paths could be found for which the end-effector would converge.

singularities are situations in which the system could converge on a target location, but the system simply cannot reach the target from its current configuration. Figure 7.6 shows an example of this phenomenon, where an obstacle would block the joints from allowing the end-effector to converge on the target. An important note, a path may exist for which the system could converge, but given its current trajectory it will never converge. The planar equivalent is reaching a dead-end within a maze: a structural singularity imposes an additional constraint on the boundary of the workspace.

Clearly, given Equation 7.20, the target location is a stable point in the field; however, this analysis was not able to conclusively determine when the system would converge on the target or when it might fail to converge due to competing forces acting on the manipulator. However, experimental results show promise that this method could be further developed to account for all scenarios, or at least conclusively determine which systems will converge.

7.3.3 Linked Robot Results

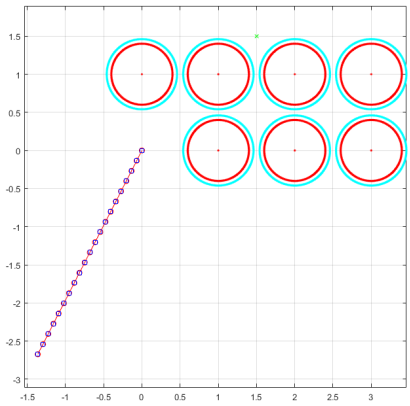
Similar to the results for the end-effector only case; a snake-like manipulator was given twenty revolute joints and was placed in a field of obstacles. In this case, the end-effector

was guided by the Secant approach potential, and the subsequent joints were subject to the TAPF approach. The end-effector was given a target location within its workspace, and each of the subsequent joints had the consecutive link as their target location: ensuring that they would never converge but that they would not hinder the end-effector from converging. The result is shown in Figure 7.7. Numerous examples of the Secant and TAPF approaches applied to manipulators could be provided, but Figure 7.7 demonstrates everything relevant with the system behavior. The high dimensional system was capable of avoiding solid bodied obstacles while converging on a target location in the task space.

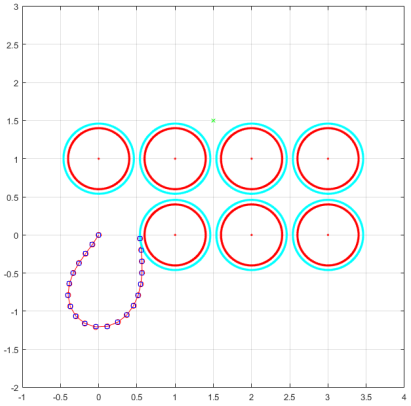
7.3.4 Experimental Results

Of course, the results for the snake-like robot can be expanded upon for real-world systems. The same algorithm was implemented for the UR5 arm produced by Universal Robots, a six degree-of-freedom manipulator. In this case, similar to the snake-like robot, the end-effector is guided by the potential gradient from the Secant approach, and the joints are repelled by obstacles dictated by the TAPF approach. For this experiment, the robot was given a desired position for the end-effector in 3D space and an obstacle of known size and position was placed in the workspace. In practical applications, a system for localizing obstacles will likely be necessary, but this experiment is simply meant to demonstrate the path planning. Capturing a system's motion in three dimensions can be challenging, so a simulation using V-REP: the "virtual robot experimentation platform" by Coppelia Robotics was conducted first for clarity, images of which are shown in Figures 7.8 and 7.9. Using the same controls, the same system was then implemented in practice, demonstrated in Figures 7.10 and 7.11. In the practical demonstration, the obstacle is replaced by a case of smaller dimension than the obstacle in the simulation and the target is a group of apples, but the principal remains the same. These figures demonstrate the capabilities of the system on practical applications.

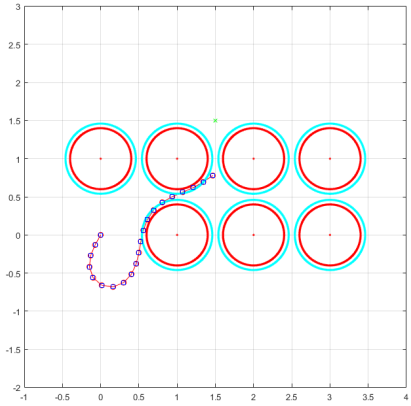
In this case, a few practical modifications have to be addressed in order to avoid col-



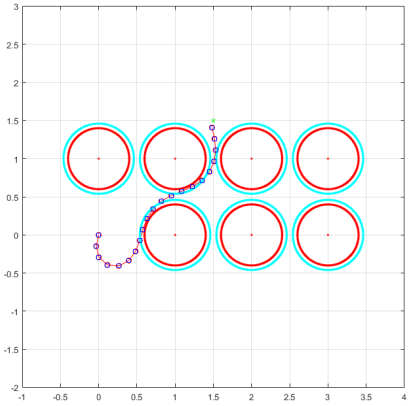
(a) Twenty link, snake-like robot, initial condition



(b) One-third through its path

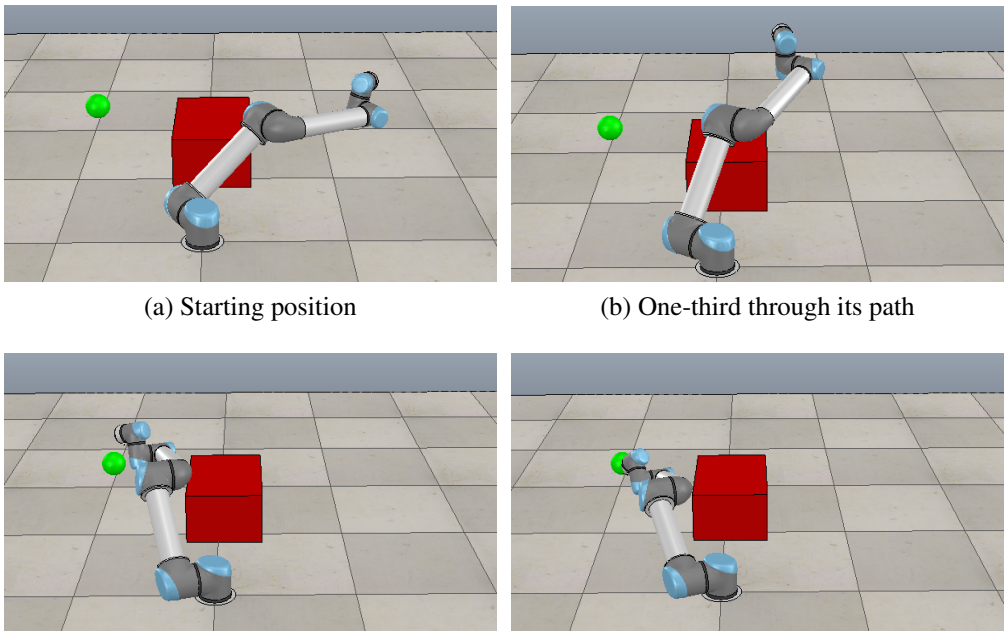


(c) Two-thirds through its path



(d) Converged on target

Figure 7.7: Demonstration of a twenty link robot arm comprised of revolute joints converging on a target location through a field of solid, circular obstacles with a fixed base located at [0, 0]. In this example, the end-effector is guided to the target by the Secant approach, and subsequent joints are avoiding obstacles based on the repulsive force from the TAPF approach.



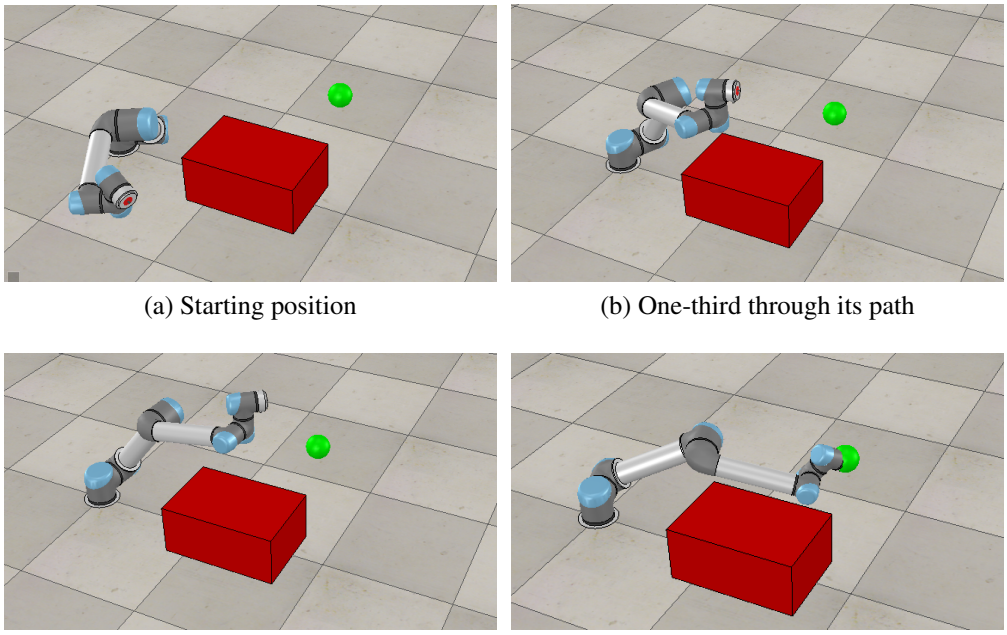
(a) Starting position

(b) One-third through its path

(c) Two-thirds through its path

(d) Converged location

Figure 7.8: A simulation using V-REP demonstrating convergence for the UR5 robot arm with a single obstacle. In this demonstration, the Secant approach is used to guide the end-effector and the repellent forces from the TAPF approach are used to avoid collisions. In this case, the red box is the obstacle and the green sphere is the target location. In this example, the arm has collided with the green sphere intentionally; the green sphere is meant as a visual aid, not an obstacle.



(a) Starting position

(b) One-third through its path

(c) Two-thirds through its path

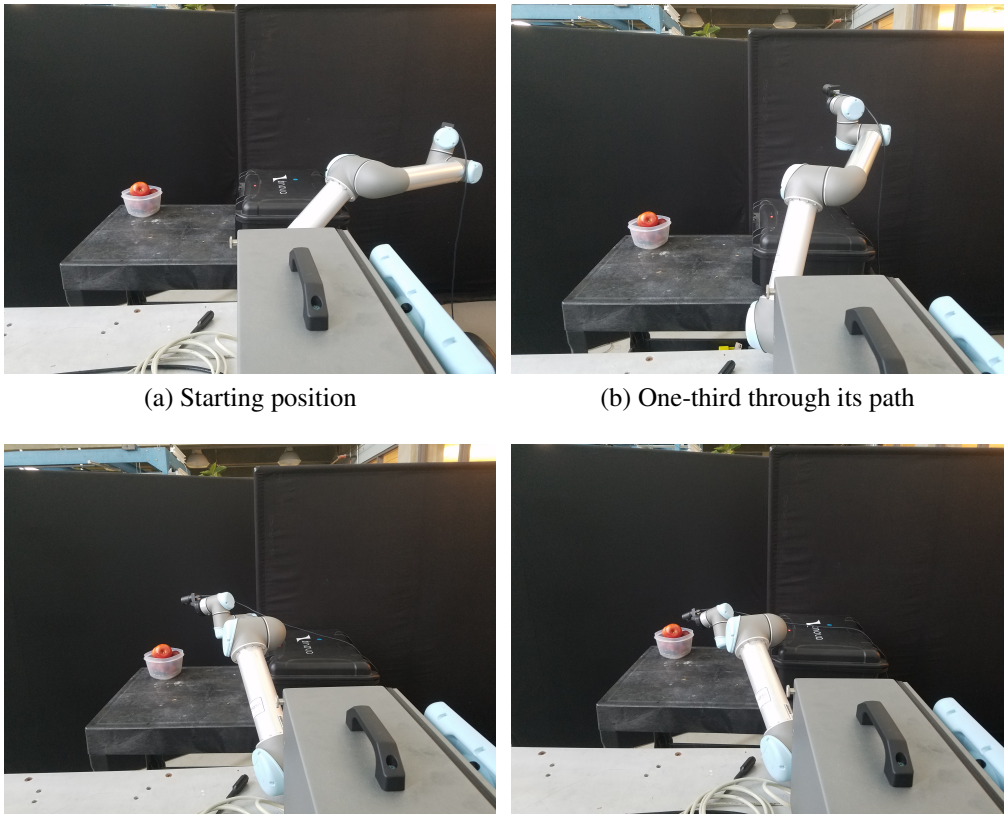
(d) Converged location

Figure 7.9: A simulation using V-REP demonstrating convergence for the UR5 robot arm with a single obstacle. This is the same simulation as shown in Figure 7.8 but from a different angle.

lisions with the obstacle. The system has physical shape which must be avoided: a point is no longer sufficient. Thus, the obstacle defined must be a bit larger than the actual obstacle to avoid intersection. Furthermore, even if the system is mathematically convergent regardless of the coefficients used, in practice, the coefficients will determine the behavior of the arm, and the physical system has boundary conditions to torque output that are not addressed in the path planning scheme. Finally, the numeric stability of the system has not been addressed in this analysis, but in practice, demonstrating stability in the numeric integration is also critical. While this simple experiment shows that the Secant and TAPF approaches can be applied to manipulators, it is only the beginning of the analysis.

7.4 Conclusion

One of the great challenges within path planning is in compensating for high degree-of-freedom manipulators. High dimensional systems are naturally challenging for most path



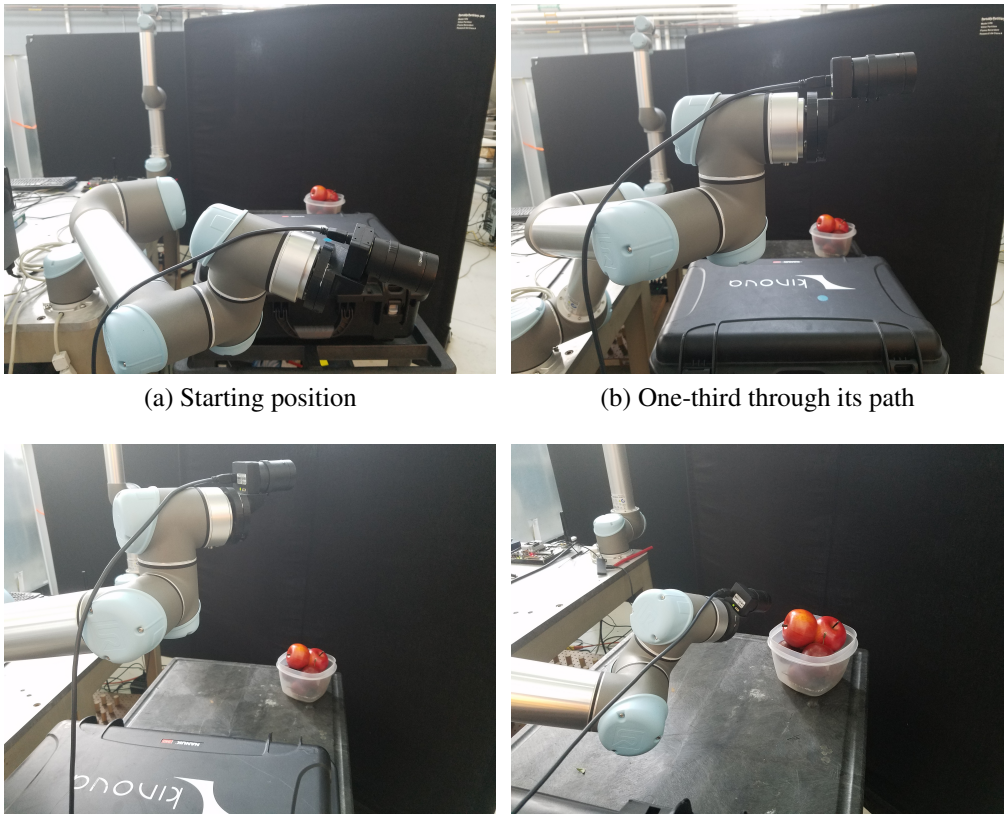
(a) Starting position

(b) One-third through its path

(c) Two-thirds through its path

(d) Converged location

Figure 7.10: The UR5 robot avoiding an obstacle (a case) in 3D space while converging on a target location (pointing towards a group of apples). The path that the arm is following is the same as the one shown in Figure 7.8.



(a) Starting position

(b) One-third through its path

(c) Two-thirds through its path

(d) Converged location

Figure 7.11: A different angle showing the convergence of the UR5 arm on a group of apples while avoiding an obstacle. This is the same routine with a different angle as shown in Figure 7.10.

planning methods and the necessary conversion between obstacles in task space and control in configuration space can seriously diminish the performance of most search methods. APF methods with guaranteed convergence properties naturally resolve both of these issues. Chapter 5 demonstrated that extremely high dimensional space for a point robot was within the purview of APF methods. Furthermore, since the artificial forces are in task space, with the target and goal location, their calculations are very simple as well. The only remaining hurtle is in bridging the gap between task space and configuration space. This transformation is naturally completed by the Jacobian transpose. While robotic arm control is often solved with the Jacobian inverse for three or six dimensional cases (where the Jacobian is square), these methods fail for redundant manipulators, high dimensional arms, or in singular configurations. The Jacobian transpose is always computable for any real-world system. Even in singular positions, as long as the gradient of the potential is not in line with the joints, the system will still move towards convergence.

The method of applying multiple frames with goal and obstacle positions for each joint is not guaranteed to converge on the target location. It is possible for the summation of forces to align with the geometry of the arm, resulting in zero net torque on the joints. Similarly, the arm may reach a virtual "dead-end" and not converge on its target location, even if viable paths exist within the field, resulting in a structural singularity. These considerations are not a result of the APF methods used themselves, but rather they arise from the application of the APF methods from the n-dimensional point robot to a general robotic manipulator. Furthermore, the convergence criteria can actually be used to benefit the system performance. Some joints may not need to converge on a target location, but they should still avoid obstacles. In this case, placing the target location in a spot where it will result in zero torque on the joint will ensure that it is both never reached and will allow for obstacle avoidance. The system may have infinite solutions that allow for convergence to the target location by the end-effector with otherwise arbitrary joint positions: convergence of all joints may not be necessary to complete real-world tasks. Only enough points nec-

essary to guarantee that the desired position and orientation of the system are reached are needed for application.

Control of robotic arms is often difficult to the uncertainty in the mass and inertia parameters of the system. As seen in Equation 7.1, convergence is not dependent on the real world parameters of the system; however, the system dynamics will affect real world performance, and the gain parameters should be tuned to achieve desired results. Overall, even if convergence is not guaranteed, the Secant and TAPF methods can be applied to manipulators with a great deal of success, as long as the limitations are of the method are understood.

CHAPTER 8
CONCLUSION

Methods for path planning in high dimensional spaces have to be generalizable and computationally efficient. Any form of search method will have its computational convergence time expand exponentially with dimension. This expansion is an inherent factor of the "search" required to find the desired path. However, APF methods do not rely on searching the space. Instead, APF methods rely on constructing a system of forces based on the gradient of a potential field to guide the system. This potential gradient is a summation attractive and repulsive forces designed to guide the robot to the target position while avoiding obstacles. This method was at one time implemented since it was extremely computationally efficient; however, the general method has convergence issues which significantly impede its usefulness. The Secant and TAPF approaches were therefore developed to combat these performance issues and guarantee convergence.

Arguably the simplest path planning problem to solve is a point robot navigating a field of point obstacles in n-dimensional space. This scenario is extremely limited in its applicability: real-world obstacles will have a region to be avoided, not a point. However, this analysis began by examining the point-obstacle case in order to understand the Secant and TAPF approaches' capabilities and limitations. From this, so long as the initial conditions of the system do not begin on an obstacle or on a "secant line" (the line connecting an obstacle with the target location), then the system will converge on the target, regardless of the layout of obstacles. This result in itself is significant: general APF theories cannot guarantee convergence in an arbitrary field of obstacles. However, by demonstrating that a positive definite Lyapunov equation has a derivative with respect to time which is negative definite, then the Secant and TAPF approaches must converge on their target locations.

Avoiding points in an n-dimensional space with a point robot is not sufficient for even

the most basic path planning approaches. In order to be the least bit applicable, the path planning approach must be able to avoid solid obstacles. From this desire, the boundary layer method was developed. The boundary layer attempts to preserve the virtues discovered for the point case of the Secant and APF approaches and apply them to solid bodied obstacles. To meet this end, the boundary layer is defined as a finite distance surrounding the obstacle. When the robot is outside of a boundary layer, then the obstacle appears as a point, and the properties are preserved. Inside the boundary layer, the space is transformed, such that the distance between the boundary layer and the radius of the obstacle is stretched such that the obstacle appears to be a point. The result is that the apparent position of the robot has shifted within the field, and the potential field warps to account for this discrepancy. By ensuring that the transition between the outside and inside of the boundary layer is smooth, then the system preserves the convergent properties of the field of point obstacles. The added conditions to convergence is that boundary layers cannot overlap and the target cannot exist within a boundary layer. However, the benefit of this method is that the obstacles do not need to have a constant radius nor do they need to be convex. If the obstacles are describable in polar coordinates, then they are accountable within this transformation. Given the boundary layer, complex shapes can be converted to point obstacles mathematically, allowing for the Secant and TAPF approaches to be an applicable path planning technique for avoiding solid obstacles.

The Secant and TAPF methods are intended to be path planning systems in n-dimensional space; however, they have been shown to be applicable in general robotic systems. Planar robots and manipulators can also be found to have convergent paths while avoiding obstacles using the Secant and TAPF approaches. However, the method by which the APF algorithms applied will change the convergent properties of the approaches. One way to apply the APF algorithm to these more complex systems is to view the structure as a series of points with their own frame of reference and goal positions. In this manner, convergence is dependent on the coupled system, each of which experiences constraints based

on the overall structure. Planar robots are reduced to a series of two degree-of-freedom points linked together with predefined target locations. Manipulators are viewed as a series of links, pinned together to form the geometry of the arm. This method still imposes restrictions on the application, and it does not guarantee convergence as broadly as the point robot scenario, but trajectory planning for generalized robots is possible with the Secant and TAPF approaches.

While a wide range of robotic applications were examined within this analysis, the Secant and TAPF approaches could be used in a multitude of path planning routines that were not mentioned. The manipulator case still needs to be proven to be convergent while avoiding obstacles for each link in the arm. Furthermore, more complex models of the manipulator could be contrived which would prohibit self-intersection. In the planar case, the rotational example could be expanded for non-holonomic vehicles. An especially exciting application for these approaches could be swarm robotics: where each independent system has a target and sees every other robot as a moving obstacle. These scenarios and more could benefit from the Secant and TAPF approaches to path planning.

As robotic systems become more complex, path planning routines will have to be able to handle high dimensional problems. Robotic manipulators, cooperative systems, hybrid systems, and swarm robotics all exist in extremely high dimensional spaces, and current path planning methods are not well equipped to solve for them. The Artificial Potential Field algorithm was developed specifically because it was computationally efficient; however, the detriments to its performance capabilities made it impracticable to use. The Secant and TAPF approaches introduce a new form of potential field for the APF algorithm which guarantee convergence for point robots, and which has been shown to be expendable to real world scenarios. Using these methods, trajectory planning in high-dimensional space is realizable due to the linear growth with dimension that APF methods exhibit. If a scenario can be pathed in real time with a given processor, and then the dimensionality of the system doubles, doubling the processing power would be sufficient to solve for the new system.

With the Secant and TAPF approaches, complex robotics in arbitrary spaces can be made to converge on targets while avoiding obstacles using a single algorithm.

REFERENCES

[1] B. Cohen, I. A. Sucan, and S. Chitta, "A Generic Infrastructure for Benchmarking Motion Planners," *IEEE International Conference on Intelligent Robots and Systems*, vol. 7, no. 12, pp. 589–595, 2012.

[2] F. Fahimi, *Autonomous Robots: Modeling, Path Planning, and Control*. Springer Science and Business Media, 2009, ISBN: 978-0-387-09537-0. arXiv: 978-0-387-09538-7 [10.1007].

[3] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, and Y. Xia, "Survey of Robot 3D Path Planning Algorithms," *Journal of Control Science and Engineering*, vol. 2016, 2016.

[4] S. LaValle, *Planning algorithms*. Cambridge University Press, 2006, p. 842, ISBN: 9780511546877. arXiv: arXiv:1011.1669v3.

[5] Kozlowski R., *Robot Motion and Control 2009*, M. Thoma, F. Allgower, and M. Morari, Eds. Springer-Verlag Berlin Heidelberg, ISBN: 9783642240638.

[6] B. Siciliano, L. Sciavico, L. Villani, and J. Oriolo, *Robotics: Modeling, Planning and Control*. Springer-Verlag London Limited, 2010, p. 644, ISBN: 9781846286414.

[7] V. Trajkovik and A. Mishev, *ICT Innovations*. New York: Springer, 2013, ISBN: 9783319014654.

[8] H. Choset, "Coverage of Known Spaces: The Boustrophedon Cellular Decomposition," *Autonomous Robots*, vol. 9, pp. 247–253, 2000.

[9] K. Hoff, T. Culver, J. Keyser, M. Lin, and D. Manocha, "Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware," *Proceedings of ACM SIGGRAPH*, pp. 277–286, 1999.

[10] L. J. Guibas, D. E. Knuth, and M. Sharir, "Randomized incremental construction of Delaunay and Voronoi diagrams," *Algorithmica*, vol. 7, no. 1-6, pp. 381–413, 1992.

[11] L. P. L. Chew, R. D. III, and R. L. S. Dyrsdale, "Voronoi diagrams based on convex distance functions," in *Proceedings of the first annual symposium on Computational geometry - SCG '85*, vol. 75, 1985, pp. 235–244, ISBN: 0897911636.

[12] J. Rosell and P. Iniguez, "Path planning using Harmonic Functions and Probabilistic Cell Decomposition," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, no. April, pp. 1803–1808, 2005.

[13] D. Hsu, J.-C. Latombe, and R. Motwani, "Path Planning in Expansive Configuration Spaces," *International Journal of Computational Geometry & Applications*, vol. 09, pp. 495–512, 1999. arXiv: arXiv:1011.1669v3.

[14] F Lingelbach, "Path planning using probabilistic cell decomposition," *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04*, vol. 1, no. April, 467–472 Vol.1, 2004.

[15] R. Bohlin, "Path Planning Using Lazy PRM," in *IEEE International Conference on Robotics & Automation*, 2000, pp. 521–528.

[16] P. C. Chen and Y. K. Hwang, "SANDROS: A Dynamic Graph Search Algorithm for Motion Planning," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 3, pp. 390–403, 1998.

[17] J. J. Kuffner and S. LaValle, "RRT-Connect : An Efficient Approach to Single-Query Path Planning," in *IEEE International Conference on Robotics & Automation*, 2000, pp. 995–1001.

[18] B. J. Cohen, G. Subramanian, S. Chitta, and M. Likhachev, "Planning for Manipulation with Adaptive Motion Primitives," in *IEEE International Conference on Robotics & Automation*, 2011, pp. 5478–5475.

[19] K. Gochev, V. Narayanan, B. Cohen, A. Safonova, and M. Likhachev, "Motion Planning for Robotic Manipulators with Independent Wrist Joints," in *IEEE International Conference on Robotics & Automation*, 2014, pp. 461–468.

[20] D. Berenson, S. S. Srinivasa, D. Ferguson, A. Collet, and J. J. Kuffner, "Manipulation planning with Workspace Goal Regions," *2009 IEEE International Conference on Robotics and Automation*, pp. 618–624, 2009.

[21] M. V. Weghe, D. Ferguson, and S. S. Srinivasa, "Randomized Path Planning for Redundant Manipulators without Inverse Kinematics," in *7th IEEE-RAS International Conference on Humanoid Robots*, 2007, pp. 477–482.

[22] M. Likhachev, G. Gordon, and S. Thrun, "ARA *: Anytime A * with Provable Bounds on Sub-Optimality,"

[23] J. Pearl, *Heuristics: Intelligent search strategies for computer problem solving*. 5. Reading: Addison-Wesley, 1985, vol. 21, p. 471, ISBN: 0201055945.

[24] N. Sadegh, "Dual objective motion planning subject to state constraints," *Robotica*, vol. 35, pp. 1–19, 2015.

[25] R. E. Korf, "Linear-Space Summary of Search: Summary Of Results," *AAAI*, pp. 533–538, 1992.

[26] B. Bonet and H. Geffner, "Planning as heuristic search," *Artificial Intelligence*, vol. 129, no. February 2000, pp. 5–33, 2001.

[27] T. Ikeda and H. Imai, "Enhanced A algorithms for multiple alignments: optimal alignments for several sequences and k-opt approximate alignments for large cases," *Theoretical Computer Science*, vol. 210, no. 2, pp. 341–374, 1999.

[28] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient Optimization Techniques for Efficient Motion Planning," in *IEEE International Conference on Robotics & Automation*, 2009, pp. 489–494.

[29] M. Zucker, N. Ratliff, a. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. a. Bagnell, and S. S. Srinivasa, "CHOMP: Covariant Hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.

[30] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4569–4574, 2011. arXiv: `arXiv:1011.1669v3`.

[31] C. Park, J. Pan, and D. Manocha, "ITOMP: Incremental trajectory optimization for real-time replanning in dynamic environments," *Proceedings of the International Conference on Automated Planning and Scheduling*, no. May 2016, pp. 207–215, 2012.

[32] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion Planning with Sequential Convex Optimization and Convex Collision Checking," *The International Journal of Robotics Research*, pp. 9–13, 2014.

[33] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 2, pp. 224–241, 1992.

[34] F. Fahimi, H. Ashrafiuon, and C. Nataraj, "Obstacle avoidance for spatial hyper-redundant manipulators using harmonic potential functions and the mode shape technique," *Journal of Robotic Systems*, vol. 20, no. 1, pp. 23–33, 2003.

[35] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *Robotics and Automation, IEEE Transactions on*, vol. 16, no. 5, pp. 615–620, 2000.

[36] E. J. Gomez, F. M. Santa, and F. H. M. Sarmiento, "A comparative study of geometric path planning methods for a mobile robot: Potential field and voronoi diagrams," *Engineering Mechatronics and Automation (CIIMA), 2013 II International Congress of*, vol. 00, no. c, pp. 1–6, 2013.

[37] T. T. Mac, C. Copot, A. Hernandez, and R. De Keyser, "Improved potential field method for unknown obstacle avoidance using UAV in indoor environment," in *SAMI 2016 - IEEE 14th International Symposium on Applied Machine Intelligence and Informatics - Proceedings*, 2016, pp. 345–350, ISBN: 9781467387408.

[38] J.-O. Kim and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *Proceedings 1991 IEEE International Conference on Robotics and Automation*, vol. 8, no. 3, pp. 338–349, 1992.

[39] Y Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," *In: Proc. IEEE International Conference on Robotics and Automation (ICRA)*, no. April, pp. 1398–1404, 1991.

[40] E. Houssein and C. Harik, "Combining Hector SLAM and Artificial Potential Field for Autonomous Navigation Inside a Greenhouse," *Robotics*, vol. 7, no. 22, 2018.

[41] B. H. Krogh and C. E. Thorpe, "Integrated path planning and dynamic steering control for autonomous vehicles," *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, vol. 3, pp. 1664–1669, 1986.

[42] A. Nooraliei, "Without local minimum by using make circular obstacles," *Proceedings - 2009 International Conference on Information Engineering and Computer Science, ICIECS 2009*, pp. 1–4, 2009.

[43] E. Rimon and D. Koditschek, "Exact Robot Navigation Using Artificial Potential Functions," *Robotics and Automation, IEEE*, vol. 8, no. 5, pp. 501–518, 1992.

[44] C. W. Warren, "Global Path Planning Using Artificial Potential Fields," *Ieee*, pp. 316–321, 1989.

[45] H. Min, Y. Lin, S. Wang, F. Wu, and X. Shen, "Path planning of mobile robot by mixing experience with modified artificial potential field method," *Advances in Mechanical Engineering*, vol. 7, no. 11, pp. 1–17, 2015.

[46] Y. Lv, X. Yang, Y. Yang, S. Pan, and C. Xin, "Formation control of UAVs based on artificial potential field," in *MATEC Web of Conferences*, vol. 10018, 2018.

[47] J. Slotine and W. Li, *Applied Nonlinear Control*, ser. Prentice-Hall International Editions. Prentice-Hall, 1991, ISBN: 9780130400499.

VITA

Konrad Ahlin, age 29, was born in 1988 in Beverly Massachusetts. Growing up in Gloucester, Konrad attended St. Johns Preparatory School in Danvers between the years of 2003 and 2007. Konrad was then accepted to Rochester Institute of Technology's Mechanical Engineering program. He earned his Bachelor of Science in 2012 with the title of Highest Honors, and then he enrolled in the Master of Science in the Mechanical Engineering program at the same institution. In 2014, he graduated the program, having successfully defended his thesis on the "Inertia Coupled Rimless Wheel". The Rimless Wheel is a mechanical walking device designed to be energy efficient, and Konrad's work adapted the system to account for practical energy losses and improved the system's overall performance. Throughout most of Konrad's time at RIT, he worked as a teachers assistant; helping with grading, monitoring, and teaching. In the last year of his Master's Program, Konrad worked as a graduate research assistant for Dr. Mario Gomes, developing the Rimless Wheel project full-time. After RIT, Konrad moved to Atlanta Georgia to study the field of Robotics through the George Woodruff School of Mechanical Engineering at Georgia Institute of Technology. During this time, he worked as a graduate research assistant under Dr. Sadegh and Dr. Hu while completing projects for the Georgia Tech Research Institute. While developing his thesis: improved Artificial Potential Field approaches to high dimensional path planning, Konrad worked on several projects involving robotic manipulators and visual servoing. Namely, he developed a method of localization using a monoscopic camera as well as a path planning routine in an unstructured environment with limited information. At present, Konrad Ahlin plans on staying with Georgia Tech Research Institute after his thesis is complete as a full time researcher.